

**PENGEMBANGAN SISTEM INFORMASI MANAJEMEN  
AKADEMIK SEKOLAH DASAR UNTUK PENDAFTARAN DAN  
PENDATAAN ULANG SISWA  
(STUDI KASUS PADA SEKOLAH DASAR NEGERI DINOYO 1  
MALANG)**

**SKRIPSI**

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun oleh:

Nyoman Puja Satria Wibawa

NIM: 155150401111027



**PROGRAM STUDI SISTEM INFORMASI  
JURUSAN SISTEM INFORMASI  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA  
MALANG  
2018**

## PENGESAHAN

### PENGEMBANGAN SISTEM INFORMASI MANAJEMEN AKADEMIK SEKOLAH DASAR UNTUK PENDAFTARAN DAN PENDATAAN ULANG SISWA (STUDI KASUS PADA SEKOLAH DASAR NEGERI DINOYO 1 MALANG)

#### SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun Oleh :

Nyoman Puja Satria Wibawa

NIM: 155150401111027

Skrripsi ini telah diuji dan dinyatakan lulus pada  
27 Desember 2018

Telah diperiksa dan disetujui oleh:

Pembimbing I

Pembimbing II

Ismiarta Aknuranda, S.T., M.Sc., Ph.D.

Satrio Agung Wicaksono, S.Kom., M.Kom.

NIK. 201006 740719 1 001

NIP. 19860521 201212 1 001

Mengetahui

Ketua Jurusan Sistem Informasi



Herwan Dille, Dr. Eng., S.T., M.T.

NIP. 19740823 200012 1 001

## PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 28 Desember 2018



Nyoman Puja Satria Wibawa  
NIM: 155150401111027

## PRAKATA

Puji syukur penulis panjatkan kehadirat Tuhan Yang Maha Esa, karena dengan rahmat-Nya penulis dapat menyelesaikan skripsi yang berjudul “PENGEMBANGAN SISTEM INFORMASI MANAJEMEN AKADEMIK SEKOLAH DASAR UNTUK PENDAFTARAN DAN PENDATAAN ULANG SISWA (STUDI KASUS PADA SEKOLAH DASAR NEGERI DINOYO 1 MALANG)”. Sejak awal sampai dengan akhir penulisan ini, sangat banyak bantuan yang penulis terima, dan karenanya dalam kesempatan ini penulis ingin menyampaikan terima kasih kepada:

1. Bapak Ismiarta Aknuranda, ST., M.Sc., Ph.D selaku dosen pembimbing I yang telah memberikan kesempatan, saran, motivasi dan waktunya untuk membimbing penulis. Serta memberikan contoh sebagai pribadi yang sabar, solutif, tekun dan professional dalam memberikan bimbingan kepada penulis.
2. Bapak Satrio Agung Wicaksono, S.Kom, M.Kom, selaku dosen pembimbing II yang telah memberikan kesempatan dan waktu untuk membimbing penulis dalam menyelesaikan penelitian, serta mengajarkan ketegasan dan kedisiplinan dalam menyelesaikan kewajiban.
3. Bapak Herman Tolle, Dr. Eng., S.T, M.T. selaku Ketua Jurusan Sistem Informasi Universitas Brawijaya Malang.
4. Bapak Yusi Tyroni Mursityo, S.Kom., M.AB. selaku Ketua Program Studi Sistem Informasi Universitas Brawijaya Malang.
5. Seluruh Bapak dan Ibu dosen beserta karyawan yang telah mendidik dan membantu penulis selama menempuh pendidikan di Fakultas Ilmu Komputer Universitas Brawijaya.
6. Bapak I Putu Oka dan Ibu Ni Putu Sukiati sebagai orang tua penulis, Mbak Putu Priska Jelita, dan Mbak Kadek Indah Maheswari sebagai saudara penulis yang senantiasa memberikan do’a dan dukungan baik dalam bentuk moril maupun materiil kepada penulis sehingga skripsi ini dapat terselesaikan. Serta sebagai motivasi kepada penulis untuk menyelesaikan skripsi.
7. Bapak Drs. H. Totok Wargo Santoso, MM selaku Kepala Sekolah Dasar Negeri Dinoyo 1 yang telah memberikan izin kepada penulis untuk melakukan penelitian skripsi pada Sekolah Dasar Negeri Dinoyo 1. Serta selalu membantu penulis dalam menyelesaikan skripsi.
8. Bapak Donny dan Ibu Tasya selaku Petugas Tata Usaha Sekolah Dasar Negeri Dinoyo 1 yang telah bersedia untuk menjadi narasumber, menguji sistem informasi yang dikembangkan serta memberikan masukan-masukan terhadap pengembangan sistem informasi ini.
9. Irvan Dk, Fikri Rizky, Edo Yokko, Arrofi Arga, Imam Safii, Nazva Abiya, Rivalno, Nabila Aulia, Goesti Sofi, Teguh Pribadi, Vira Indriana, Irfanda, dan Chaerul Irfan sebagai sahabat Eyang Brotherhood yang selalu membantu penulis dari awal masuk perkuliahan, serta memberikan canda tawa,

semangat, motivasi dan dukungan kepada penulis untuk menyelesaikan penelitian skripsi.

10. Falih Farhan, Wira Kumara, Qomarul Umam, Abid, Alfath, Putu Pradnyana, Raka Yuwono sebagai sahabat Sarjana Sukses yang selalu memberikan motivasi, semangat, canda tawa, dan membantu penulis dalam menyelesaikan penelitian skripsi ini.
11. Kadek Saka, Jeri Pemana, Satya Krisna, Dwi Aryasa, Adit Widiartha, dan Mayoga Putra selaku teman kontrakan Neptunus yang selalu memberikan canda tawa, semangat, motivasi, bantuan dan selalu menemani penulis dalam menyelesaikan penelitian skripsi.
12. Teman-teman Lembaga Badan Perwakilan Mahasiswa Sistem Informasi, Eksekutif Mahasiswa Sistem Informasi, dan Departemen Penelitian Pengembangan Studi yang telah menjadi sarana bagi penulis untuk berkembang menjadi pribadi yang solutif, peka terhadap permasalahan dan memberikan kesempatan untuk aktif berorganisasi di kalangan mahasiswa dan masyarakat.
13. Teman-teman mahasiswa Jurusan Sistem Informasi dan jurusan lainnya dari Fakultas Ilmu Komputer yang tidak dapat penulis sebutkan namanya satu per satu, yang telah menjadi teman yang baik bagi penulis, senantiasa memberikan dukungan terhadap sistem informasi yang dikembangkan serta dukungan moril dan motivasi, sehingga mampu terus menyemangati penulis untuk segera menyelesaikan penelitian yang dikerjakan.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih banyak kekurangan baik format laporan maupun isinya. Oleh karena itu, penulis mengharapkan saran dan kritik yang membangun dari para pembaca guna perbaikan dalam penelitian yang selanjutnya. Semoga skripsi ini dapat bermanfaat baik bagi pembaca maupun penulis, amin.

Malang, 28 Desember 2018

Penulis

satriacap@gmail.com



## ABSTRAK

**Nyoman Puja Satria Wibawa, Pengembangan Sistem Informasi Manajemen Akademik Sekolah Dasar Untuk Pendaftaran Dan Pendataan Ulang Siswa (Studi Kasus Pada Sekolah Dasar Negeri Dinoyo 1 Malang)**

**Pembimbing: Ismiarta Aknuranda, ST., M.Sc., Ph.D dan Satrio Agung Wicaksono, S.Kom., M.Kom**

Sekolah Dasar Negeri (SDN) Dinoyo 1 merupakan salah satu instansi pendidikan dasar yang berada di daerah Malang. SDN Dinoyo 1 telah lulus 8 Standar Nasional Pendidikan dan memiliki akreditasi A. Pada saat ini proses pendaftaran siswa baru masih dilakukan secara manual sehingga menyebabkan permasalahan dalam mengelola data pendaftaran. Setiap satu semester SDN Dinoyo 1 melakukan proses pendataan ulang data pokok setiap siswa. Proses pendataan ulang data pokok siswa saat ini juga dilakukan secara manual yang menyebabkan pihak sekolah susah dalam pengelolaan data pokok siswa tersebut. Maka dari itu diperlukan sistem informasi yang dapat menangani masalah tersebut. Pengembangan sistem informasi, dilakukan dengan menggunakan *Waterfall model* dan diimplementasikan pada situs web. Pada tahap analisis persyaratan, dihasilkan 44 persyaratan fungsional dan 1 persyaratan nonfungsional. Pada tahap perancangan, dihasilkan 26 kelas pada diagram kelas analisis, 5 kelas pada diagram kelas perancangan *controller*, dan 4 kelas pada diagram kelas perancangan *model*. Tahap implementasi menggunakan kerangka kerja *Codeigniter* dengan pola *Model-View-Controller*. Kemudian untuk menguji hasil implementasi dari sistem dilakukan pengujian validasi pada beberapa fungsi. Hasil dari pengujian validasi mendapatkan hasil valid. Pengujian juga dilakukan dengan menggunakan pengujian kompatibilitas peramban. Hasil pengujian kompatibilitas peramban menunjukkan sistem berjalan dengan *Critical Issues* pada jenis peramban firefox versi 63 dan Chrome versi 70, Kemudian terdapat *Major issues* apabila situs diakses melalui peramban Internet Explorer versi 11 dan Andriod versi 4 atau sebelumnya.

Kata kunci: Sistem Informasi, *Waterfall model*, Pengujian validasi, *Codeingiter*, situs web

## ABSTRACT

**Nyoman Puja Satria Wibawa, *Development Of Education Management Information System In Elementary School For Student Registration And Reverification***

**Supervisors: Ismiarta Aknuranda, ST., M.Sc., Ph.D dan Satrio Agung Wicaksono, S.Kom., M.Kom**

*Dinoyo 1 Elementary School is one of the primary educational institutions in Malang city. Dinoyo 1 Elementary School has passed 8 National Standards of Education and has earned 'A' accreditation. In the process of registering new students it is still done manually, causing problems in managing registration data. Every semester, Dinoyo 1 elementary school re-updating data for each student that is also done manually which causes the school committee experiencing difficulties in managing the student's data. Therefore, it is necessary to have an information system that can handle these problems. In developing the information system, it is done by using the Waterfall model and implemented in a website platform. At the requirements analysis phase, 44 functional requirements and 1 nonfunctional requirement are obtained. At the design phase, 26 classes in the class analysis diagram, 5 classes in the controller design class diagram, and 4 classes in the model design class diagram are obtained. The implementation phase is done by using CodeIgniter framework with the Model-View-Controller pattern. Afterward, the results of the implementation of the system is tested by validation testing on several functions. The results of validation testing are valid. The testing is also done using compatibility testing. The results of browser compatibility testing show that the system runs with Critical Issues on Firefox browser version 63 and Chrome version 70, then there are major issues if the site is accessed via Internet Explorer version 11 and Android version 4 browsers or earlier.*

**Keywords:** *Information System, Waterfall model, Validation testing, CodeIgniter, website*

## DAFTAR ISI

<b>PENGESAHAN .....</b>	<b>i</b>
<b>PERNYATAAN ORISINALITAS .....</b>	<b>ii</b>
<b>PRAKATA .....</b>	<b>iii</b>
<b>ABSTRAK.....</b>	<b>v</b>
<b>ABSTRACT .....</b>	<b>vi</b>
<b>DAFTAR ISI.....</b>	<b>vii</b>
<b>DAFTAR TABEL.....</b>	<b>xi</b>
<b>DAFTAR GAMBAR .....</b>	<b>xiv</b>
<b>DAFTAR LAMPIRAN.....</b>	<b>xvi</b>
<b>BAB 1 PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	3
1.3 Tujuan .....	4
1.4 Manfaat.....	4
1.5 Batasan masalah .....	4
1.6 Sistematika pembahasan.....	4
<b>BAB 2 LANDASAN KEPUSTAKAAN .....</b>	<b>6</b>
2.1 Kajian Pustaka .....	6
2.2 Sekolah Dasar.....	7
2.2.1 Sekolah Dasar Negeri Dinoyo 1.....	7
2.3 Sistem Informasi .....	8
2.3.1 Sistem Informasi Manajemen .....	9
2.4 <i>Software Development Life Cycle</i> .....	10
2.4.1 <i>Waterfall Model</i> .....	10
2.5 Tahap Definisi Kebutuhan.....	11
2.5.1 Proses Bisnis.....	12
2.5.2 <i>Business Process Model and Notation (BPMN)</i> .....	13
2.5.3 Pemodelan <i>Use case</i> .....	18
2.6 Tahap Perancangan Sistem.....	22
2.6.1 <i>Unified Modelling Language</i> .....	22
2.6.2 <i>Physical Data Model</i> .....	28



2.7 Tahap Implementasi Sistem.....	29
2.7.1 <i>Object Oriented Programming (OOP)</i> .....	29
2.7.2 <i>Model-View-Controller (MVC)</i> .....	29
2.8 Tahap Pengujian Sistem.....	30
2.8.1 <i>Black-Box Testing</i> .....	30
<b>BAB 3 METODOLOGI .....</b>	<b>32</b>
3.1 Wawancara .....	32
3.2 Studi Pustaka.....	33
3.3 Analisis Persyaratan .....	33
3.4 Perancangan Sistem.....	33
3.5 Implementasi Sistem .....	34
3.6 Pengujian Sistem.....	34
3.7 Kesimpulan Dan Saran .....	34
<b>BAB 4 ANALISIS PERSYARATAN .....</b>	<b>35</b>
4.1 Aturan Penomoran .....	35
4.2 Pemodelan Proses Bisnis .....	36
4.2.1 Analisis Proses Bisnis <i>As-Is</i> .....	36
4.2.2 Proses Bisnis <i>To-Be</i> .....	45
4.3 Analisis Persyaratan .....	51
4.3.1 Identifikasi Tipe Pemangku Kepentingan.....	51
4.3.2 Identifikasi Kebutuhan Pemangku Kepentingan dan Pengguna ..	51
4.3.3 Identifikasi Pengguna .....	54
4.3.4 Identifikasi Fitur .....	55
4.3.5 Persyaratan Fungsional .....	57
4.3.6 Persyaratan Nonfungsional.....	60
4.4 Pemodelan <i>Use case</i> .....	61
4.4.1 <i>Use Case Diagram</i> .....	61
4.4.2 Deskripsi Aktor .....	63
4.4.3 Spesifikasi <i>Use case</i> .....	64
<b>BAB 5 PERANCANGAN .....</b>	<b>75</b>
5.1 Perancangan Arsitektur Sistem.....	75
5.2 Pemodelan Objek.....	75

5.2.1 Diagram Kelas Analisis.....	76
5.2.2 Diagram Kelas Perancangan.....	78
5.3 Pemodelan Interaksi Objek Perancangan.....	82
5.3.1 Memasukkan Data Pendaftaran Siswa Baru .....	82
5.3.2 Menampilkan Detail Data Pokok Siswa .....	83
5.3.3 Verifikasi Fomulir dan Berkas Pendaftaran.....	84
5.3.4 Memperbaharui Data Pokok Siswa .....	85
5.4 Perancangan Basis Data .....	86
5.5 Perancangan Algoritme .....	88
5.5.1 Menampilkan Formulir Pendaftaran Siswa Baru .....	88
5.5.2 Memasukkan Data Pendaftaran Siswa Baru .....	89
5.5.3 Menampilkan Detail Data Pokok Siswa .....	90
5.5.4 Memperbaharui Data Pokok Siswa.....	91
5.6 Perancangan Antarmuka Pengguna .....	92
5.6.1 Antarmuka Pengguna <i>Login</i> .....	92
5.6.2 Antarmuka Pengguna Memasukkan Data Pendaftaran Siswa Baru .....	92
5.6.3 Antarmuka Pengguna Menampilkan Data Pokok Siswa .....	93
5.6.4 Antarmuka Pengguna Memperbaharui Data Pokok Siswa .....	94
<b>BAB 6 IMPLEMENTASI .....</b>	<b>96</b>
6.1 Spesifikasi Lingkungan Implementasi .....	96
6.2 Struktur Artefak Sistem .....	97
6.3 Implementasi Algoritme .....	98
6.3.1 Menampilkan Formulir Pendaftaran Siswa Baru .....	99
6.3.2 Memasukkan Data Pendaftaran Siswa Baru .....	99
6.3.3 Menampilkan Detail Data Pokok Siswa .....	104
6.3.4 Memperbaharui Data Pokok Siswa .....	104
6.4 Implementasi Antarmuka Pengguna .....	109
6.4.1 Antarmuka Pengguna <i>Login</i> .....	109
6.4.2 Antarmuka Pengguna Memasukkan Data Pendaftaran Siswa Baru .....	109
6.4.3 Antarmuka Pengguna Menampilkan Data Pokok Siswa .....	110
6.4.4 Antarmuka Pengguna Memperbaharui Data Pokok Siswa .....	111

<b>BAB 7 PENGUJIAN .....</b>	<b>112</b>
7.1 Pengujian Validasi .....	112
7.1.1 Pengujian Validasi Memasukkan Data Pendaftaran Siswa Baru .....	112
7.1.2 Pengujian Validasi Mengelola Data Pokok Siswa .....	114
7.2 Pengujian Kompatibilitas Peramban .....	116
<b>BAB 8 PENUTUP .....</b>	<b>117</b>
8.1 Kesimpulan .....	117
8.2 Saran .....	118
<b>DAFTAR REFERENSI .....</b>	<b>119</b>
<b>LAMPIRAN A RENCANA WAWANCARA .....</b>	<b>121</b>
A.1 Rencana Wawancara Kepala SDN Dinoyo 1 .....	121
A.2 Rencana Wawancara Petugas Tata Usaha SDN Dinoyo 1 .....	121
<b>LAMPIRAN B HASIL WAWANCARA .....</b>	<b>122</b>
B.1 Wawancara Kepala SDN Dinoyo 1 .....	122
B.2 Wawancara Petugas Tata Usaha SDN Dinoyo 1 .....	124
<b>LAMPIRAN C HASIL PENGUJIAN VALIDASI .....</b>	<b>126</b>
C.1 Hasil Pengujian Validasi Menampilkan Data Pokok Siswa .....	126
C.2 Hasil Pengujian Validasi Merubah Data Pokok Siswa .....	127

## DAFTAR TABEL

Tabel 2.1 Notasi <i>Events</i> .....	14
Tabel 2.2 Notasi <i>Activity</i> .....	14
Tabel 2.3 Notasi <i>Gateways</i> .....	15
Tabel 2.4 Notasi Data .....	16
Tabel 2.5 Notasi <i>Artifacts</i> .....	17
Tabel 2.6 Notasi <i>Connecting Object</i> .....	17
Tabel 2.7 Notasi <i>Swimlaness</i> .....	18
Tabel 2.8 Analisis Tipe Pemangku Kepentingan .....	19
Tabel 2.9 Kerangka Dokumentasi Pernyataan Masalah .....	20
Tabel 2.10 Contoh Tabel Fitur .....	20
Tabel 2.11 Format Dokumentasi Spesifikasi <i>Use case</i> .....	21
Tabel 2.12 Notasi <i>Use case</i> .....	23
Tabel 2.13 Notasi <i>Sequence Diagram</i> .....	24
Tabel 2.14 Notasi <i>Sequence Diagram</i> (lanjutan) .....	25
Tabel 2.15 Notasi <i>Class Diagram</i> .....	26
Tabel 4.1 Realisasi Aturan Penomoran .....	35
Tabel 4.2 Identifikasi Aktor pada Proses Bisnis.....	36
Tabel 4.3 Hasil Identifikasi Proses Bisnis As-Is .....	37
Tabel 4.4 Daftar Aktivitas Proses Bisnis As-Is Pendaftaran Siswa Baru .....	38
Tabel 4.5 Daftar Aktivitas Proses Bisnis As-Is Pendataan Ulang Data Pokok Siswa .....	41
Tabel 4.6 Identifikasi Permasalahan .....	42
Tabel 4.7 <i>Problem Statement 1</i> .....	44
Tabel 4.8 <i>Problem Statement 2</i> .....	44
Tabel 4.9 Hasil Identifikasi Proses Bisnis <i>To-Be</i> .....	46
Tabel 4.10 Daftar Aktivitas Proses Bisnis <i>To-Be</i> Pendaftaran Siswa Baru .....	47
Tabel 4.11 Perubahan Aktivitas Proses Bisnis Pendaftaran Siswa Baru .....	48
Tabel 4.12 Daftar Aktivitas Proses Bisnis <i>To-Be</i> Pendataan Ulang Data Pokok Siswa .....	49
Tabel 4.13 Perubahan Aktivitas Proses Bisnis Pendataan Ulang Data Pokok.....	50
Tabel 4.14 Analisis Tipe Pemangku Kepentingan .....	51

Tabel 4.15 Hasil Identifikasi Kebutuhan Pengguna.....	52
Tabel 4.16 Hubungan Kebutuhan Pengguna dengan Analisis Masalah.....	54
Tabel 4.17 Hasil Identifikasi Pengguna .....	55
Tabel 4.18 Hasil Identifikasi Fitur .....	55
Tabel 4.19 Hubungan Kebutuhan Pengguna dengan Fitur .....	57
Tabel 4.20 Persyaratan Fungsional .....	57
Tabel 4.21 Persyaratan Nonfungsional .....	60
Tabel 4.22 Hubungan Aktivitas Proses Bisnis <i>To-Be</i> dengan <i>Use case</i> .....	62
Tabel 4.23 Hubungan <i>Use case</i> dengan Fitur.....	63
Tabel 4.24 Deskripsi Aktor .....	64
Tabel 4.25 Spesifikasi <i>Use case Login</i> .....	64
Tabel 4.26 Spesifikasi <i>Use Case</i> Memasukkan Data Pendaftaran Siswa Baru .....	65
Tabel 4.27 Spesifikasi <i>Use case</i> Verifikasi Formulir dan Berkas Pendaftaran.....	67
Tabel 4.28 Spesifikasi <i>Use case</i> Mengelola Data Pendaftaran Siswa Baru .....	68
Tabel 4.29 Spesifikasi <i>Use Case</i> Memonitoring Proses Seleksi.....	70
Tabel 4.30 Spesifikasi <i>Use Case</i> Memasukkan Data Pokok Siswa.....	71
Tabel 4.31 <i>Spesifikasi Use case</i> Mengelola Data Pokok Siswa.....	72
Tabel 5.1 <i>Pseudocode</i> Menampilkan Formulir Pendaftaran Siswa Baru .....	88
Tabel 5.2 <i>Pseudocode</i> Memasukkan Data Pendaftaran Siswa Baru.....	89
Tabel 5.3 <i>Pseudocode</i> Memasukkan Data Pendaftaran Siswa Baru.....	90
Tabel 5.4 <i>Pseudocode</i> Memperbaharui Data Pokok Siswa.....	91
Tabel 6.1 Spesifikasi Perangkat Keras .....	96
Tabel 6.2 Spesifikasi Perangkat Lunak .....	96
Tabel 6.3 Spesifikasi Minimal Lingkungan <i>Deployment</i> .....	97
Tabel 6.4 Implementasi Algoritme Menampilkan Formulir Pendaftaran Siswa Baru .....	99
Tabel 6.5 Implementasi Algoritme Memasukkan Data Pendaftaran Siswa Baru .	99
Tabel 6.6 Implementasi Algoritme Menampilkan Detail Data Pokok Siswa.....	104
Tabel 6.7 Implementasi Algoritme Memperbaharui Data Pokok Siswa .....	105
Tabel 7.1 Matriks Skenario Pengujian Validasi Memasukkan Data Pendaftaran Siswa Baru .....	112
Tabel 7.2 Rencana Pengujian dan Kasus Uji Fungsi Memasukkan Data Pendaftaran Siswa Baru .....	113



Tabel 7.3 Rencana Pengujian dan Kasus Uji Fungsi Menangani Tidak Lengkapnya Data .....	113
Tabel 7.4 Matriks Skenario Pengujian Validasi Mengelola Data Pokok Siswa ....	114
Tabel 7.5 Rencana Pengujian dan Kasus Uji Fungsi Menampilkan Detail Data Pokok Siswa.....	115
Tabel 7.6 Rencana Pengujian dan Kasus Uji Fungsi Merubah Data Pokok Siswa	115



## DAFTAR GAMBAR

Gambar 2.1 Aktivitas Dasar Sistem Informasi.....	8
Gambar 2.2 Klasifikasi Tipe Sistem Informasi .....	9
Gambar 2.3 <i>Waterfall Model</i> .....	10
Gambar 2.4 Pemodelan Artefak Sistem.....	27
Gambar 2.5 Pemodelan <i>Package</i> .....	28
Gambar 2.6 Contoh <i>Physical Data Model</i> .....	29
Gambar 2.7 Struktur MVC.....	30
Gambar 3.1 Diagram Alur Penelitian .....	32
Gambar 4.1 Proses Bisnis <i>As-Is</i> Pendaftaran Siswa Baru .....	38
Gambar 4.2 Proses Bisnis <i>As-Is</i> Pendataan Ulang Data Pokok Siswa .....	40
Gambar 4.3 Proses Bisnis <i>To-Be</i> Pendaftaran Siswa Baru .....	46
Gambar 4.4 Proses Bisnis <i>To-Be</i> Pendataan Ulang Data Pokok Siswa.....	49
Gambar 4.5 <i>Use case Diagram</i> .....	62
Gambar 5.1 Arsitektur Sistem Informasi Manajemen Akademik .....	75
Gambar 5.2 Diagram Kelas Analisis.....	76
Gambar 5.3 Analisis Atribut .....	77
Gambar 5.4 Diagram Kelas Untuk <i>Controller</i> .....	78
Gambar 5.5 Diagram Kelas Perancangan Untuk <i>Model</i> .....	79
Gambar 5.6 Diagram Kelas Untuk <i>View</i> .....	80
Gambar 5.7 Relasi Diagram Kelas <i>View</i> dan <i>Controller</i> .....	81
Gambar 5.8 Diagram Kelas Relasi Subkelas <i>Controller</i> dan <i>Model</i> .....	82
Gambar 5.9 <i>Sequence Diagram</i> Memasukkan Data Pendaftaran Siswa Baru.....	83
Gambar 5.10 <i>Sequence Diagram</i> Menampilkan Detail Data Pokok Siswa .....	84
Gambar 5.11 <i>Sequence Diagram</i> Verifikasi Formulir dan Berkas Pendaftaran ....	85
Gambar 5.12 <i>Sequence Diagram</i> Memperbaharui Data Pendataan Ulang Data Pokok Siswa.....	86
Gambar 5.13 <i>Physical Data Model</i> .....	87
Gambar 5.14 Antarmuka Pengguna <i>Login</i> .....	92
Gambar 5.15 Antarmuka Pengguna Pendaftaran Siswa .....	93
Gambar 5.16 Antarmuka Pengguna Pengelolaan Data Pendaftar .....	94
Gambar 5.17 Antarmuka Pengguna Formulir Pendataan Ulang .....	95

Gambar 6.1 Implementasi Diagram Kelas untuk <i>View</i> .....	97
Gambar 6.2 Struktur Artefak Sistem .....	98
Gambar 6.3 Antarmuka Pengguna <i>Login</i> .....	109
Gambar 6.4 Antarmuka Pengguna Memasukkan Data Pendaftaran Siswa Baru	110
Gambar 6.5 Antarmuka Pengguna Menampilkan Data Pokok Siswa .....	111
Gambar 6.6 Antarmuka Pengguna Memperbaharui Data Pokok Siswa .....	111
Gambar 7.1 Kodifikasi Pengujian Validasi .....	112
Gambar 7.2 Kategori Kompatibilitas Pada Aplikasi SortSite .....	116
Gambar 7.3 Hasil Pengujian Kompatibilitas Peramban .....	116



## DAFTAR LAMPIRAN

<b>LAMPIRAN A RENCANA WAWANCARA .....</b>	<b>121</b>
A.1 Rencana Wawancara Kepala SDN Dinoyo 1 .....	121
A.2 Rencana Wawancara Petugas Tata Usaha SDN Dinoyo 1 .....	121
<b>LAMPIRAN B HASIL WAWANCARA .....</b>	<b>122</b>
B.1 Wawancara Kepala SDN Dinoyo 1 .....	122
B.2 Wawancara Petugas Tata Usaha SDN Dinoyo 1 .....	124
<b>LAMPIRAN C HASIL PENGUJIAN VALIDASI .....</b>	<b>126</b>
C.1 Hasil Pengujian Validasi Menampilkan Data Pokok Siswa.....	126
C.2 Hasil Pengujian Validasi Merubah Data Pokok Siswa .....	127



## BAB 1 PENDAHULUAN

### 1.1 Latar belakang

Menurut Undang-Undang Nomor 20 Tahun 2003, dinyatakan bahwa pendidikan merupakan usaha menciptakan suasana belajar yang terencana agar peserta didik dapat mengembangkan potensi dirinya. Salah satu pendidikan formal di Indonesia adalah pendidikan dasar. Sekolah memiliki peran sebagai instansi pendidikan formal untuk mencari ilmu dan kegiatan belajar mengajar (Suharjo, 2006). Dengan perkembangan teknologi informasi saat ini, sekolah berusaha untuk memberikan pelayanan yang baik dalam bidang pendidikan dan juga menyebarluaskan informasi kepada masyarakat umum. Sekolah Dasar Negeri (SDN) Dinoyo 1 merupakan salah satu instansi pendidikan dasar yang berada di daerah Malang. SDN Dinoyo 1 telah lulus 8 Standar Nasional Pendidikan dan memiliki akreditasi A.

Berdasarkan hasil wawancara dengan Bapak Totok Wargo sebagai kepala SDN Dinoyo 1 dan Ibu Tasya sebagai Kepala Tata Usaha pada Lampiran B.1 dan Lampiran B.2, dalam melaksanakan kegiatan pendaftaran siswa baru masih dilakukan secara manual. Pengumuman dan informasi pendaftaran siswa baru ditempel pada papan pengumuman dan spanduk sehingga orang tua calon siswa harus datang ke sekolah untuk mengetahui informasi tersebut. Pada proses pendaftaran, orang tua calon siswa mengisi buku tamu dan mengambil formulir berupa kertas di sekolah. Setelah formulir telah diisi, orang tua calon siswa mengembalikan formulir pendaftaran beserta berkas-berkas pendukung seperti *fotocopy* akte kelahiran dan ijazah Taman Kanak-Kanak kepada pihak sekolah untuk dilakukan verifikasi. Dalam proses verifikasi, pihak sekolah akan mencocokkan data yang telah diisikan pada formulir dengan data yang ada pada *fotocopy* akte kelahiran dan ijazah Taman Kanak-Kanak. Setelah proses pendaftaran selesai, maka pihak sekolah akan menentukan siswa yang dinyatakan diterima sebagai siswa SDN 1 Dinoyo. Kemudian pihak sekolah akan mencatat data siswa yang diterima berdasarkan formulir dan berkas-berkas pendukung tersebut dengan menggunakan *Microsoft Excel*, data ini akan menjadi data pokok siswa. Data pokok siswa berguna sebagai identitas resmi setiap siswa.

Pada proses pendaftaran siswa baru, terjadi permasalahan yaitu (1) terkadang pengisian formulir pendaftaran oleh orang tua siswa tidak lengkap sehingga data yang dibutuhkan pihak sekolah kurang lengkap; (2) Formulir pendaftaran yang ditulis secara manual membuat pihak sekolah terkadang sulit untuk membaca data pada formulir untuk dilakukan verifikasi; (3) Permasalahan lain muncul ketika kurang lengkapnya pengumpulan berkas-berkas pendukung oleh orang tua calon siswa. 4) Pihak sekolah juga kesusahan dalam membandingkan data umur siswa dan juga jarak rumah ke sekolah dari setiap siswa untuk menentukan siswa yang diterima; (5) setelah data siswa diterima, petugas tata usaha harus memasukkan semua data pada formulir secara manual, proses tersebut membutuhkan waktu yang lama dan juga petugas tata usaha kesusahan jika ada data yang tidak lengkap.



Ibu Tasya juga mengatakan setiap akhir semester pihak sekolah akan melaksanakan pendataan ulang terkait data pokok setiap siswa. Pendataan ulang berguna untuk memperbarui data pokok jika ada perubahan di setiap semester dan data pokok yang telah diperbaharui akan dilaporkan kepada Dinas Pendidikan Malang. Proses pendataan ulang masih dilakukan secara manual, yaitu setiap guru kelas akan membagikan formulir pendataan ulang kepada setiap siswa, kemudian orang tua siswa akan mengisi formulir pendataan ulang dan selanjutnya akan diberikan kembali kepada guru kelas. Pihak sekolah kemudian membandingkan setiap data yang ada pada formulir pendataan ulang dengan data yang telah disimpan pada *Microsoft Excel*. Jika ada perubahan, maka petugas tata usaha memperbaharui data pokok siswa tersebut. Pada proses pendataan ulang juga terjadi permasalahan yaitu (1) Petugas Tata Usaha mengalami kesulitan dalam membandingkan setiap data pada formulir pendataan ulang yang telah diisi orang tua siswa dengan data yang sudah ada, serta membutuhkan waktu yang lama; (2) tidak lengkapnya data yang diisikan pada formulir; (3) tulisan yang susah dibaca; (4) data pokok siswa yang berupa *Microsoft Excel* juga susah diakses dari mana saja, dan petugas tata usaha mengalami kesulitan jika ingin mengelola data pokok siswa tersebut.

Proses pendaftaran siswa baru dan pendataan ulang siswa yang dilakukan secara manual menyebabkan pihak sekolah melakukan dua kali kerja dalam hal pencatatan data. Selain itu data yang belum terpusat menyebabkan pihak sekolah kesulitan dalam mengelola data karena hanya dapat diakses dari satu tempat saja.

Sistem informasi manajemen merupakan salah satu teknologi informasi yang memiliki peran penting dalam bidang akademik dan administrasi di sekolah. Kemampuan guru dan staf sekolah dalam bekerja menggunakan sistem informasi manajemen telah berkembang dan mampu untuk mendukung berbagai kegiatan administrasi di sekolah (Shah, 2013). Berdasarkan permasalahan yang muncul pada proses pendaftaran siswa baru dan pendataan ulang, diperlukan alat bantu yang memiliki fungsi untuk mengelola data pada formulir pendaftaran dan data pada formulir pendataan ulang yang terpusat untuk membantu pihak sekolah dan pihak siswa dalam mengakses data tersebut. Penelitian ini bertujuan untuk mengembangkan sebuah alat bantu berupa sistem informasi manajemen akademik yang dapat membantu permasalahan yang telah dijelaskan

Untuk mengembangkan sebuah sistem informasi, terdapat model konseptual yang digunakan sebagai acuan yaitu model *Software Development Life Cycle* (SDLC). Salah satu dari model SDLC adalah metode *Waterfall*. Pada prinsipnya metode *waterfall* harus merencanakan dan menjadwalkan semua proses kegiatan sebelum mulai bekerja. Dengan begitu maka perangkat lunak yang akan dibangun akan terencana, sesuai kebutuhan dan jika ada kebutuhan baru maka akan dilakukan pada tahap kelima pada *waterfall* model. *Waterfall* model cocok digunakan pada penelitian ini karena proses bisnis mengenai pendaftaran siswa baru dan pendataan ulang data pokok siswa secara manual sudah ditetapkan dari pihak sekolah, sehingga kebutuhan dan persyaratan sistem selalu tetap dan telah dipahami dengan baik (Sommerville, 2011).

Tahap pertama dalam metode *waterfall* adalah melakukan analisis dan definisi kebutuhan melalui konsultasi dengan pengguna sistem. Tahap kedua adalah dengan melakukan desain sistem untuk merencanakan perangkat lunak yang melibatkan identifikasi dan penggambaran secara fundamental sistem perangkat lunak dan hubungannya. Tahap ketiga adalah dengan melakukan implementasi dan pengujian unit dengan melibatkan verifikasi setiap unit dan spesifikasinya. Tahap keempat adalah integrasi dan pengujian sistem untuk memastikan bahwa perangkat lunak sudah sesuai dengan kebutuhan pengujian ini dilakukan dengan pengguna sistem. Tahap kelima adalah operasi dan pemeliharaan untuk meningkatkan kualitas sistem saat adanya persyaratan baru (Sommerville, 2011).

Penelitian ini menggunakan pendekatan berorientasi objek karena memiliki beberapa keuntungan yaitu bersifat *reusable* karena obyek digunakan dalam suatu masalah dapat dipakai ulang, kemudahan dalam pemeliharaan karena menggunakan obyek yang dapat diubah jika terdapat perubahan, dan dengan pewarisan dapat meningkatkan konsistensi dalam kode program (Shalahuddin & S, 2015). Pada tahap pengujian peneliti menggunakan *Validation Testing* sebagai pengujian yang fokus pada tindakan dan hasil dari sistem yang terlihat oleh pengguna dan pengujian kompatibilitas peramban yang berguna untuk mengetahui apakah aplikasi memiliki masalah ketika berjalan pada komputer, tampilan perangkat, sistem operasi, browser dan kecepatan koneksi jaringan yang berbeda. (Pressman, 2010).

Penelitian sebelumnya dilakukan oleh Jawara Wahyu Al Faraday pada tahun 2018, dengan masalah yang melatarbelakangi yaitu proses pendataan hasil belajar siswa yang masih konvensional dan juga proses penerimaan siswa baru yang masih konvensional dengan menggunakan formulir. Penelitian tersebut menjelaskan bahwa sistem informasi manajemen dapat dikembangkan pada Sekolah Dasar untuk memudahkan proses pendaftaran siswa baru. Penelitian berikutnya dilakukan oleh Madiha Shah tahun 2013 yang membahas mengenai dampak adanya sistem informasi manajemen pada sekolah. Penelitian tersebut menjelaskan bahwa kemampuan guru dan staf saat ini dapat mendukung kegiatan administrasi di sekolah.

Berdasarkan uraian yang telah dijelaskan, peneliti tertarik untuk melakukan penelitian terkait dengan judul “Pengembangan Sistem Informasi Manajemen Akademik Sekolah Dasar Untuk Pendaftaran dan Pendataan Ulang Siswa (Studi Kasus Pada Sekolah Dasar Negeri Dinoyo 1 Malang)”.

## 1.2 Rumusan masalah

Berdasarkan uraian latar belakang yang disebutkan, maka dapat dirumuskan permasalahan dalam penelitian ini sebagai berikut:

- 1) Bagaimana hasil analisis persyaratan sistem informasi manajemen akademik pada Sekolah Dasar Negeri Dinoyo 1?
- 2) Bagaimana hasil rancangan sistem informasi manajemen akademik pada Sekolah Dasar Negeri Dinoyo 1?

- 3) Bagaimana hasil implementasi sistem informasi manajemen akademik pada Sekolah Dasar Negeri Dinoyo 1 berdasarkan hasil identifikasi persyaratan dan perancangan sistem?
- 4) Bagaimana hasil pengujian sistem informasi manajemen akademik pada Sekolah Dasar Negeri Dinoyo 1?

### 1.3 Tujuan

Tujuan yang ingin dicapai dalam penelitian ini adalah:

- 1) Mengidentifikasi persyaratan sistem informasi manajemen akademik pada Sekolah Dasar Negeri Dinoyo 1.
- 2) Merancang sistem informasi manajemen akademik pada Sekolah Dasar Negeri Dinoyo 1.
- 3) Mengimplementasi sistem informasi manajemen akademik pada Sekolah Dasar Negeri Dinoyo 1 berdasarkan hasil identifikasi persyaratan dan perancangan sistem.
- 4) Menguji sistem informasi manajemen akademik pada Sekolah Dasar Negeri Dinoyo 1.

### 1.4 Manfaat

Manfaat dari penelitian ini adalah terwujudnya sistem informasi berbasis situs web yang digunakan oleh Sekolah Dasar Negeri Dinoyo 1 untuk mempermudah proses pendaftaran siswa baru dan pendataan ulang data pokok siswa.

### 1.5 Batasan masalah

Berdasarkan latar belakang yang telah dijabarkan sebelumnya, maka batasan masalah untuk membatasi permasalahan agar tidak melebar melebihi pokok permasalahan antara lain adalah:

- 1) Penelitian ini berfokus pada proses pendaftaran siswa baru dan pendataan ulang siswa pada Sekolah Dasar Negeri Dinoyo 1.
- 2) Metode yang digunakan dalam membangun sistem informasi adalah metode *waterfall*.
- 3) Sistem informasi yang dikembangkan pada penelitian ini hanya dilakukan pada fase *requirement analysis and definition, system and software design, implementation and unit testing, integration and system testing* pada *waterfall* model.

### 1.6 Sistematika pembahasan

Penelitian ini dibagi menjadi tujuh bab yang dibagi tiap pokok bahasan. Adapun sistematika dari penelitian ini sebagai berikut:

**BAB 1 PENDAHULUAN**

Bab ini membahas mengenai latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah, dan sistematika pembahasan penelitian.

**BAB 2 LANDASAN KEPUSTAKAAN**

Bab ini menguraikan dasar-dasar teori dan referensi yang berkaitan dengan penelitian ini.

**BAB 3 METODOLOGI**

Bab ini membahas mengenai metode dan prosedur yang digunakan dalam melakukan penelitian.

**BAB 4 ANALISIS PERSYARATAN**

Bab ini membahas mengenai persyaratan sistem yang disesuaikan dengan metode yang digunakan. Analisis persyaratan terdiri dari pemodelan proses bisnis, identifikasi tipe pemangku kepentingan, analisis masalah, identifikasi kebutuhan pemangku kepentingan dan pengguna, identifikasi pengguna, identifikasi fitur, pemodelan *use case*.

**BAB 5 PERANCANGAN**

Bab ini menjelaskan bagaimana memodelkan kebutuhan ke dalam rancangan berupa diagram perancangan sistem sesuai dengan metode yang digunakan. Diagram yang dijelaskan dalam bab perancangan di antaranya adalah *sequence diagram* dan diagram kelas, *physical data model*, perancangan algoritma, dan antarmuka pengguna.

**BAB 6 IMPLEMENTASI**

Bab ini menjelaskan mengenai spesifikasi lingkungan implementasi, beberapa hasil implementasi algoritme, dan hasil implementasi antarmuka pengguna.

**BAB 7 PENGUJIAN**

Bab ini membahas mengenai pengujian terhadap hasil analisis dan perancangan sistem yang telah dibuat berdasarkan pengujian *black box – validation testing* dan metode pengujian *kompabilitas*.

**BAB 8 PENUTUP**

Bab ini membahas mengenai kesimpulan dari keseluruhan penelitian dan saran yang diberikan berdasarkan penelitian yang sudah dilakukan.



## BAB 2 LANDASAN KEPUSTAKAAN

### 2.1 Kajian Pustaka

Penelitian ini dilakukan berdasarkan penelitian yang telah dilakukan sebelumnya. Penelitian pertama dilakukan oleh Madiha Shah tahun 2013 dengan judul *"Impact of Management Information Systems (MIS) on School Administration : What the Literature Says"*. Penelitian ini membahas dampak dari adanya sistem informasi manajemen di sekolah. Penelitian tersebut menjelaskan bahwa sistem informasi manajemen berperan penting dalam bidang pendidikan dan administrasinya. Kemampuan guru dan staf dalam bekerja dengan menggunakan teknologi informasi telah berkembang secara signifikan dan mampu untuk mendukung berbagai kegiatan administrasi di sekolah. Namun setiap sekolah memiliki kebutuhan sendiri dalam melaksanakan administrasi di sekolah, sehingga sistem informasi manajemen di setiap sekolah akan berbeda. Penelitian ini juga menjelaskan untuk membuat sistem informasi manajemen yang dapat digunakan secara efektif, dibutuhkan perancangan melalui proses induktif yang mencakup semua pemangku kepentingan pada sekolah. Kemudian setelah sistem informasi manajemen telah diterapkan, perlunya pelatihan agar setiap pengguna dapat dengan efektif menggunakan sistem tersebut.

Penelitian selanjutnya dilakukan oleh Adel Alshamrani dan Abdullah Bahattab pada tahun 2015, dengan judul *"A Comparison Between Three SDLC Models Waterfall Model, Spiral Model, and Incremental/Iterative Model"*. Pada penelitian tersebut menjelaskan bahwa *software development life cycle (SDLC)* memiliki banyak model, setiap model memiliki kelebihan dan kekurangannya tersendiri. Sehingga perlunya memiliki metode SDLC yang sesuai dengan kebutuhan dan permasalahan yang berlaku pada organisasi. Penelitian ini juga menjelaskan bahwa metode *Waterfall* cocok digunakan jika proses bisnis pada organisasi telah ditetapkan sehingga persyaratan dan kebutuhan sistem telah dipahami dengan baik. *Waterfall* model juga memiliki kelebihan yaitu di setiap fase akan diproses dan diselesaikan dalam periode waktu tertentu sehingga penjadwalan setiap fase akan jelas, mudah diterapkan banyak digunakan dan dikenal.

Penelitian yang dilakukan oleh Jawara Wahyu Al Faraday pada tahun 2018, yang membahas mengenai pembangunan sistem informasi manajemen akademik di sekolah dasar. Masalah yang melatarbelakangi yaitu penyampaian informasi dari sekolah kepada masyarakat yang masih menggunakan cara konvensional. Kemudian proses pendaftaran murid masih dilakukan secara manual dengan mengisi formulir. Proses pendataan hasil belajar siswa yang masih konvensional sehingga siswa dan orangtua harus menunggu nilai keluar dan juga proses penerimaan siswa baru yang masih secara konvensional dengan menggunakan formulir, dan proses pendataan alumni dari sekolah dasar tersebut yang sulit dilakukan. Hasil penelitian ini adalah sebuah sistem informasi manajemen akademik untuk mengatasi permasalahan tersebut. Keterkaitan penelitian



tersebut dengan penelitian yang dilakukan penulis merupakan dalam bidang dan permasalahan yang sama yaitu manajemen akademik sekolah dasar.

Penelitian selanjutnya dilakukan oleh Danniar Reza Firdausy pada tahun 2018 dengan judul “Pengembangan Sistem Informasi Manajemen Pelaporan Sarana dan Prasarana Studi Pada Fakultas Ilmu Komputer Universitas Brawijaya”. Masalah yang melatarbelakangi adalah proses pelaporan keluhan masih dilakukan dengan cara manual dengan melaporkan permasalahan ke loket perlengkapan di lantai 6 Gedung Fakultas Ilmu Komputer. Hal ini dapat menyebabkan proses membuat laporan keluhan sarana dan prasarana menjadi lebih rumit dan membutuhkan waktu yang lama. Penelitian tersebut berfokus pada pengembangan sistem informasi untuk pelaporan sarana dan prasarana dengan menggunakan metode *waterfall*. Metode *waterfall* digunakan karena kebutuhan dan persyaratan sistem telah dipahami dengan baik.

## 2.2 Sekolah Dasar

Menurut Undang-Undang Nomer 20 Tahun 2003, menyatakan bahwa pendidikan merupakan usaha yang terencana untuk menciptakan suasana belajar agar peserta didik dapat mengembangkan potensi dirinya. Pendidikan dasar merupakan prasyarat dasar untuk jenjang pendidikan yang lebih tinggi.

Sekolah sebagai instansi pendidikan formal untuk mencari ilmu dan kegiatan belajar mengajar. Pada dasarnya, sekolah dasar merupakan lembaga pendidikan yang menyelenggarakan program pendidikan selama enam tahun untuk usia anak 6 sampai 12 tahun (Suharjo, 2006).

### 2.2.1 Sekolah Dasar Negeri Dinoyo 1

Sekolah Dasar Negeri Dinoyo 1 Malang merupakan instansi pendidikan dasar yang terletak di Kecamatan Dinoyo, Kabupaten Malang dan dimiliki oleh negara. Menurut Keputusan Kepala Sekolah nomor 800, SDN Negeri 1 Dinoyo memiliki 11 Guru matapelajaran, 2 staff Tata Usaha, dan 3 Guru sebagai Bimbingan Konseling.

SDN Dinoyo 1 dinyatakan telah lulus 8 Standar Nasional yakni Standari Isi, Standar Proses, Standar Kelulusan, Standar Tenaga Pendidik, Standar Sarana Prasarana, Standar, Pengelolaan, Standar Pembiayaan, Standar Penilaian pada tahun 2016. Sehingga membuat SDN Dinoyo 1 memiliki akreditasi A.

SDN Dinoyo 1 memiliki visi yaitu :

“UNGGUL DALAM BIDANG IPTEK, PATRIOTISME, SENI DAN BUDAYA LINGKUNGAN HIDUP BERDASARKAN IMTAQ”

Serta memiliki Misi :

- 1) Membina insan yang bertaqwa kepada Tuhan Yang Maha Esa
- 2) Menyiapkan pribadi unggul baik akademik maupun non akademik
- 3) Mengembangkan insan yang cinta tanah air dan bangsa
- 4) Mengembangkan insan yang cinta budaya bangsa

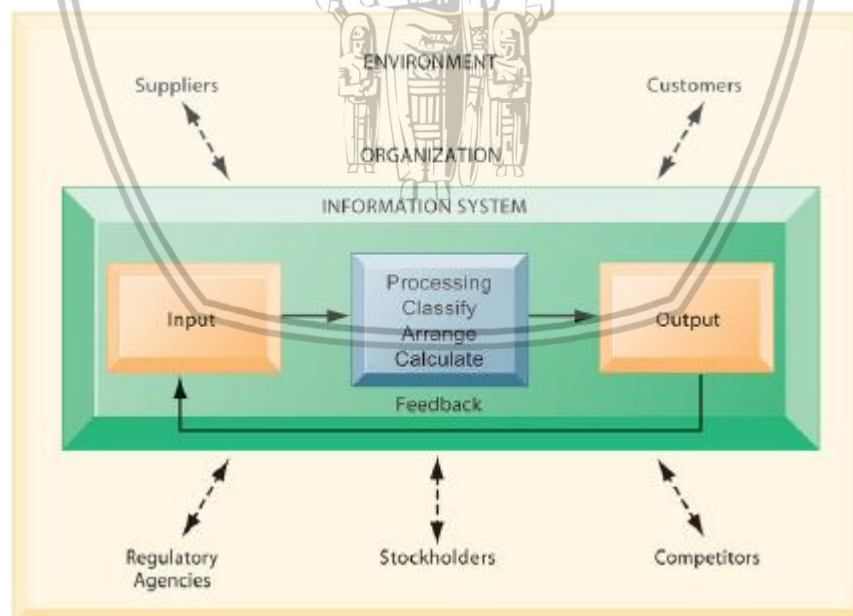
- 5) Mengembangkan pendidikan yang berbudaya lingkungan hidup.

SDN Dinoyo 1 setiap satu tahun sekali melakukan proses penerimaan siswa baru. Calon siswa harus melakukan pendaftaran pada waktu yang telah ditentukan oleh pihak sekolah, kemudian pihak sekolah akan melakukan seleksi calon siswa tersebut. Selain penerimaan siswa baru, SDN Dinoyo 1 setiap satu semester sekali melakukan pendataan ulang data identitas diri setiap siswanya. Hal ini dilakukan untuk memperbaharui data identitas setiap siswa.

## 2.3 Sistem Informasi

Sistem diartikan sebagai sekumpulan komponen yang saling berhubungan dengan batasan yang didefinisikan secara jelas untuk mencapai tujuan umumnya. Suatu sistem dapat dibuat dari sistem lain atau menjadi bagian dari sistem yang lebih besar. Sedangkan sistem informasi merupakan kumpulan dari orang, perangkat keras, perangkat lunak, jaringan komunikasi, sumber data, kebijakan dan prosedur untuk mengambil, menyimpan, mengelola dan menyebarkan informasi dalam organisasi (O'Brien & Marakas, 2013).

Sistem informasi dapat membantu manajer dan pekerja untuk membantu menyelesaikan masalah dan memvisualisasikan subjek yang kompleks. Sistem informasi memiliki informasi yang berkaitan dengan suatu objek dalam organisasi atau lingkup tertentu. Sistem informasi memiliki fungsi untuk manajemen dan mengorganisir suatu informasi yang berkaitan dengan organisasi tertentu agar informasi dapat dengan mudah diperoleh dan dikelola (Laudon & Laudon, 2011).



**Gambar 2.1 Aktivitas Dasar Sistem Informasi**

Sumber: Laudon & Laudon (2011)

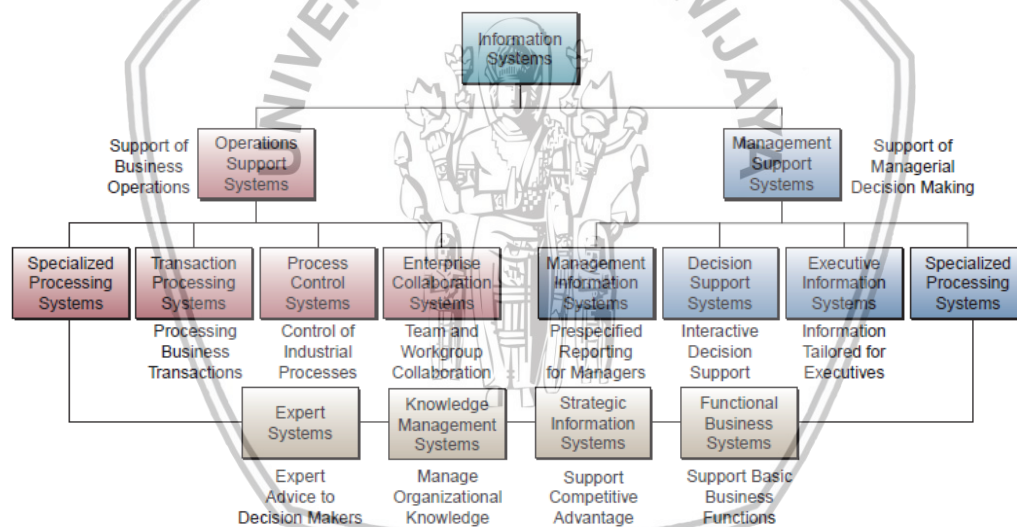
Gambar 2.1 menggambarkan aktivitas dasar sistem informasi. Sistem informasi memiliki informasi yang dimiliki organisasi beserta lingkungannya. Terdapat tiga aktivitas dasar yang dimiliki oleh sebuah sistem informasi, yaitu

*input*, proses, dan *output*. Kemudian *feedback* merupakan keluaran yang dikembalikan kepada orang atau aktivitas tertentu dalam organisasi untuk memperbaiki masukan. Aktivitas dasar ini memproduksi informasi yang dibutuhkan organisasi. Kemudian faktor lingkungan dapat mempengaruhi organisasi dan sistem informasinya seperti pelanggan, *competitor*, *stakeholder*, penyedia, dan regulasi (Laudon & Laudon, 2011).

Penerapan sistem informasi dalam suatu organisasi dapat meningkatkan efisiensi proses bisnis karena informasi dapat diakses secara cepat dan akurat. Informasi yang diperoleh dapat dikelola lebih cepat dengan adanya sistem informasi dalam organisasi. Salah satu tipe dari sistem informasi adalah sistem informasi manajemen.

### 2.3.1 Sistem Informasi Manajemen

Sistem informasi manajemen adalah sistem informasi berbasis komputer untuk merencanakan, mengontrol, membantu membuat keputusan, dan memecahkan masalah yang berkaitan dengan aktivitas dan proses bisnis suatu perusahaan (Oz, 2009).



**Gambar 2.2 Klasifikasi Tipe Sistem Informasi**

Sumber: O'Brien & Marakas (2013)

Gambar 2.2 menggambarkan klasifikasi tipe sistem informasi. Sistem informasi memiliki dua klasifikasi yaitu sebagai *operations support system* dan *management support systems*. *Management support systems* berfokus pada penyediaan informasi dan dukungan untuk mengambil keputusan yang efektif oleh manajer dan professional bisnis lainnya (O'Brien & Marakas, 2013).

Sistem informasi manajemen dapat memberikan informasi dalam bentuk laporan yang kemudian akan diberikan kepada manajer dan professional bisnis. Contohnya manajer penjualan dapat menerima hasil penjualan produk harian

dengan menggunakan jaringan komputer dan *browser* web secara *real-time* (O'Brien & Marakas, 2013).

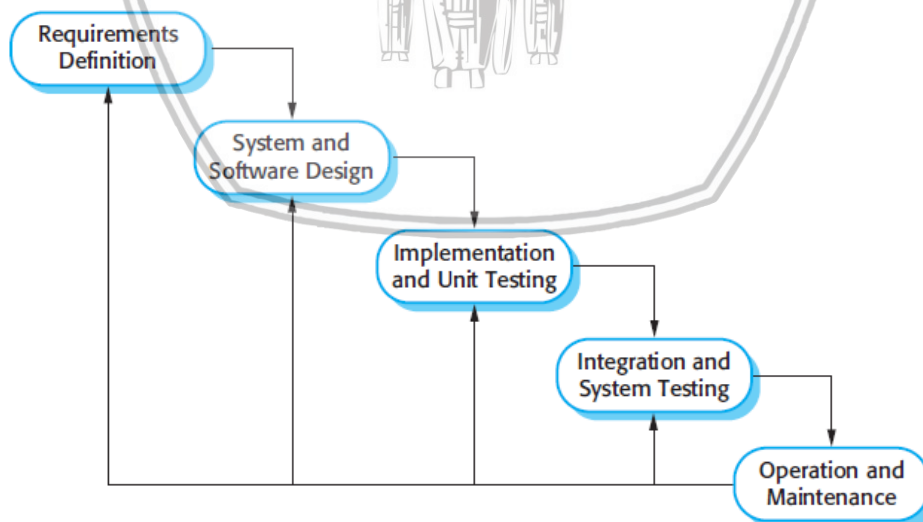
Dalam membangun dan mengembangkan sistem informasi manajemen yang mampu memenuhi kebutuhan pengguna, dibutuhkan pengetahuan mengenai proses daur hidup dari pengembangan sistem. Proses daur hidup untuk pengembangan sistem, dapat disebut sebagai proses *Software Development Life Cycle*.

## 2.4 Software Development Life Cycle

*Software development life cycle* (SDLC) adalah metode untuk mengembangkan perangkat lunak secara sistematis dan dapat meningkatkan penyelesaian proyek perangkat lunak dalam waktu dan menjaga kualitas produk perangkat lunak sesuai standar. SDLC menyediakan urutan kegiatan dalam merancang perangkat lunak yang dapat memungkinkan perusahaan untuk mengatur jadwal kerja secara efisien untuk membangun perangkat lunak dengan kerangka waktu dan anggaran tertentu (Mishra & Dubey, 2013). Dalam SDLC terdapat beberapa teknik pengembangan sistem yang umum digunakan, sedangkan untuk penelitian ini akan menggunakan teknik *Waterfall Model*.

### 2.4.1 Waterfall Model

Dari berbagai *framework* SDLC, salah satunya adalah *Waterfall Model*. *Waterfall Model* merupakan metode untuk mengembangkan perangkat lunak yang digerakkan dengan rencana. Dalam setiap kegiatan, wajib melakukan perencanaan dan menjadwalkan terlebih dahulu sebelum memulai mengerjakan (Sommerville, 2011).



**Gambar 2.3 Waterfall Model**

Sumber: Sommerville (2011)



*Waterfall Model* bersifat linear, proses pengembangan tidak akan berlanjut ke tahap berikutnya apabila fase sebelumnya belum selesai, oleh karena itu *Waterfall Model* cocok digunakan jika persyaratan telah dipahami terlebih dahulu dan persyaratan tidak bersifat dinamis ((Mishra & Dubey, 2013).

Gambar 2.3 menjelaskan siklus hidup dalam *Waterfall Model*, berikut merupakan penjelasannya (Sommerville, 2011) :

1. *Requirements Definition* : fase untuk mendefinisikan kebutuhan, batasan, layanan dan tujuan dalam sistem. Pendefinisian dilakukan secara rinci dan berfungsi sebagai spesifikasi sistem.
2. *System and Software Design* : fase untuk mengalokasikan kebutuhan sistem dengan membangun keseluruhan arsitektur sistem. Fase ini menjelaskan identifikasi, abstraksi dan hubungan dari setiap komponen sistem.
3. *Implementation and Unit Testing* : fase untuk merealisasikan perancangan sistem sebagai satu set program atau unit program. Kemudian dilakukan pengujian unit untuk mengetahui setiap unit program apakah telah sesuai dengan kebutuhan.
4. *Integration and System Testing* : fase untuk mengintegrasikan setiap unit program dan diuji sebagai sistem yang lengkap untuk memastikan setiap kebutuhan telah terpenuhi. Kemudian perangkat lunak diberikan kepada *customer*.
5. *Operation and Maintenance* : merupakan fase terpanjang karena sistem yang telah diinstal dan digunakan akan selalu diberikan perawatan untuk memperbaiki kesalahan yang tidak ditemukan pada fase awal.

Setiap fase akan menghasilkan dokumen yang telah disetujui. Dokumen akan menjadi informasi yang dibutuhkan untuk tahap selanjutnya. Dokumen yang dihasilkan pada setiap fase memungkinkan untuk dimodifikasi jika mengalami perubahan (Sommerville, 2011). Tahapan yang digunakan dalam *Waterfall Model* akan dijelaskan dalam beberapa sub-bab selanjutnya.

## 2.5 Tahap Definisi Kebutuhan

Tahap definisi kebutuhan atau *requirements definition* merupakan tahap pertama dalam metode *waterfall* untuk mendefinisikan kebutuhan, batasan, layanan, dan tujuan dalam sistem (Sommerville, 2011). Pada penelitian ini tahap awal definisi kebutuhan dilakukan dengan melakukan analisis proses bisnis. Analisis proses bisnis akan menghasilkan pemodelan proses bisnis saat ini (*as-is*) dan pemodelan proses bisnis usulan (*to-be*) yang akan dimodelkan dengan *Business Process Model and Notation*. Kemudian hasil dari pemodelan proses bisnis *to-be* digunakan sebagai informasi dalam pemodelan *use case*. Pemodelan *use case* dilakukan identifikasi tipe pemangku kepentingan, analisis masalah, identifikasi kebutuhan pengguna, identifikasi fitur, identifikasi persyaratan fungsional dan non fungsional, pemodelan *use case*.



### 2.5.1 Proses Bisnis

Proses bisnis merupakan serangkaian aktivitas-aktivitas yang memerlukan *input* dan menghasilkan *output* sesuai kebutuhan. Sekumpulan aktivitas tersebut dilakukan secara terkoordinasi untuk mencapai tujuan organisasi. Proses bisnis dapat dilakukan oleh satu organisasi, atau bahkan lebih untuk melakukan suatu interaksi antara organisasi satu dengan organisasi lainnya (Weske, 2007)

Berikut jenis aktivitas dalam proses bisnis (Weske, 2007) :

1. Aktivitas manual merupakan aktivitas yang tidak didukung oleh sistem.
2. Aktivitas interaksi pengguna merupakan aktivitas yang dilakukan atau dikerjakan oleh karyawan menggunakan sistem informasi, sehingga membutuhkan antarmuka sistem yang baik agar pekerjaan dapat lebih efektif.
3. Aktivitas sistem, merupakan kegiatan yang tidak melibatkan peran pengguna. Semua kegiatan dilakukan oleh sistem.

Proses bisnis dapat menggambarkan model proses bisnis dalam sebuah diagram. Diagram proses bisnis akan merepresentasikan aktivitas dan tugas dari proses bisnis dan hubungan diantara keduanya. Notasi-notasi proses bisnis contohnya adalah UML *Activity Diagram* dan BPMN (Weske, 2007). Proses bisnis dapat digunakan untuk memahami sebuah organisasi atau perusahaan. Beberapa pendekatan yang dapat dilakukan untuk mengetahui proses bisnis yang ada pada organisasi adalah dengan melakukan analisis dan pemodelan terhadap proses bisnis *as-is* dan proses bisnis *to-be*.

#### 2.5.1.1 Analisis Pemodelan Proses Bisnis AS-IS

Analisis dan pemodelan proses bisnis *as-is* merupakan kegiatan untuk mengumpulkan dan memodelkan proses bisnis yang aktual pada organisasi. Hal ini dilakukan untuk menyajikan informasi mengenai situasi dan proses bisnis aktual perusahaan atau organisasi, mengetahui kekurangan dari proses bisnis dan memunculkan peningkatan dalam proses bisnis yang sudah ada (Becker, et al., 2013).

Proses bisnis *as-is* dapat diidentifikasi dengan melakukan wawancara dengan pegawai yang berpartisipasi dalam eksekusi proses bisnis. Selain itu, proses bisnis *as-is* dapat diidentifikasi dengan melakukan observasi terhadap pegawai yang sedang mengerjakan tugas-tugasnya dalam organisasi atau perusahaan (Przybytek, 2011).

#### 2.5.1.2 Analisis Pemodelan Proses Bisnis To-Be

Analisis dan pemodelan proses bisnis *to-be* merupakan kegiatan untuk mengidentifikasi dan memodelkan proses bisnis yang sudah mengalami peningkatan (*improvement*) berdasarkan analisis terhadap proses bisnis *as-is*. Penerapan proses bisnis *to-be* dapat dilakukan dalam jangka waktu menengah atau jangka panjang. (Becker, et al., 2013).

Analisis dan pemodelan proses bisnis *to-be* dilakukan dengan melibatkan pemangku kepentingan untuk dapat memperoleh kemungkinan peningkatan proses bisnis pada organisasi atau perusahaan. Peningkatan proses bisnis mungkin dilakukan dengan menerapkan otomasi pada beberapa proses bisnis. Namun, otomasi tidak menjamin peningkatan yang signifikan, sehingga perlu melibatkan pemangku kepentingan dalam kegiatan analisis dan pemodelan proses bisnis *to-be* (Przybyłek, 2011).

### 2.5.2 Business Process Model and Notation (BPMN)

BPMN merupakan kumpulan notasi grafis untuk memodelkan dan menggambarkan proses bisnis. Tujuan utama BPMN untuk memberikan penjelasan berupa gambar notasi yang lebih mudah dipahami oleh semua *business users* (Weske, 2007).

BPMN memiliki lima kategori elemen dasar, yaitu (Weske, 2007):

1. *Flow Object* : Merupakan elemen utama yang menggambarkan perilaku proses bisnis, terdiri dari *events*, *activity*, *gateways*.
2. *Artefacts* : Berguna untuk menggambarkan informasi tambahan mengenai proses bisnis. terdiri dari *data object*, *groups*, dan *text annotations*. Setiap *artefacts* dapat dikaitkan dengan *flow elements*.
3. *Connectin Object* : Berguna untuk menghubungkan flow object, swimlanes, atau artefacts. terdiri dari *sequence flows*, *message flows*, *association*, dan *data associations*.
4. *Swimlanes* : Terdiri dari *lanes* dan *Pools*. *Pools* menggambarkan organisasi yang berinteraksi terhadap beberapa proses bisnis, *pools* akan menggambarkan satu organisasi. *Lanes* merepresentasikan entitas dalam organisasi.

Object Management Group (OMG) menjelaskan bahwa BPMN dirancang untuk memberikan mekanisme pemodelan proses bisnis secara sederhana dan mudah dipahami, namun di saat yang bersamaan, BPMN juga mampu menangani visualisasi kompleksitas proses bisnis. Oleh karena itu, BPMN menyediakan sekumpulan notasi yang dibagi menjadi 5 kategori di antaranya adalah *flow objects*, *data*, *connecting objects*, *swimlanes*, *artifacts*.




#### 2.5.2.1 Flow Object

*Flow Object* menjelaskan perilaku yang dimiliki proses bisnis. Terdiri dari :

1. *Events*

*Events* merupakan notasi yang merepresentasikan *trigger* dan *results* pada sebuah proses bisnis sehingga memberikan pengaruh terhadap alur aktivitas BPMN. Penjelasan notasi terdapat pada Tabel 2.1.

**Tabel 2.1 Notasi Events**

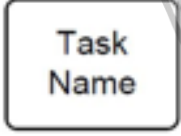
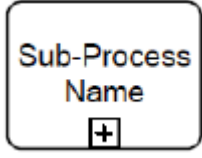
Notasi	Nama	Deskripsi
	<i>Start Event</i>	Notasi yang digunakan untuk memulai proses
	<i>Intermediate Events</i>	Notasi yang digunakan untuk menggambarkan kejadian yang ada di awal dan akhir proses. Mempengaruhi aliran proses tetapi tidak memulai atau mengakhiri proses. Notasi ini memiliki beberapa macam varian untuk merepresentasikan event yang berbeda-beda. Setiap varian di inidikasikan dengan penggunaan simbol-simbol yang berbeda di tengah notasi ini.
	<i>End</i>	Notasi yang digunakan untuk mengakhiri proses.



Sumber: Object Management Group (2011)

## 2. Activity

*Activity* merupakan notasi yang merepresentasikan sebuah pekerjaan di dalam sebuah proses. *Activity* memiliki 2 tipe notasi. Penjelasan notasi terdapat pada Tabel 2.2.

**Tabel 2.2 Notasi Activity**

Notasi	Nama	Deskripsi
	<i>Task</i>	Notasi yang merepresentasikan aktivitas yang bersifat atomic. Digunakan jika pekerjaan tidak dipecah lagi.
	<i>Sub-process</i>	Notasi yang digunakan untuk merepresentasikan sekumpulan aktivitas. <i>Sub-process</i> memiliki detail yang berisi sekumpulan aktivitas di dalamnya.





Notasi	Nama	Deskripsi
<p><b>Sequential</b></p>  <p><b>Parallel</b></p> 	<i>Multiple Instance Activity</i>	Jenis kegiatan yang bertindak sebagai pembungkus untuk kegiatan yang memiliki beberapa kejadian yang dibuat secara paralel ( <i>multiple instance activity parallel</i> ) atau berurutan ( <i>multiple instance sequential</i> ). Jika bersifat berurutan ( <i>sequential</i> ), maka kegiatan akan dikerjakan setelah kegiatan sebelumnya telah selesai. Jika bersifat paralel, maka kegiatan dikerjakan secara bersamaan ( <i>paralel</i> ).


Sumber: Object Management Group (2011)

### 3. Gateways

*Gateways* merupakan notasi digunakan untuk menjelaskan alur proses bisnis sesuai kondisi yang berjalan. Penjelasan notasi terdapat pada Tabel 2.3.

**Tabel 2.3 Notasi Gateways**

Notasi	Nama	Deskripsi
	<i>Exclusive</i>	Notasi yang menggambarkan pemecahan karena ada kondisi atau penggabungan antara beberapa kondisi. Digunakan untuk membuat alur alternative pada proses bisnis. hanya 1 alur alternatif yang dapat dijalankan sesuai dengan kondisi yang terpenuhi.
	<i>Parallel</i>	Notasi yang digunakan untuk membuat alur paralel dan menyinkronkan alur paralel yang dibuat.
	<i>Event-based</i>	Notani yang digunakan untuk alur alternative pada proses yang dipicu oleh sebuah <i>events</i> tertentu.
	<i>Complex</i>	Notasi yang digunakan untuk merepresentasikan alur kompleks dalam proses bisnis.






Notasi	Nama	Deskripsi
	<i>Inclusive</i>	Notasi yang digunakan untuk membuat alur alternative secara parallel pada proses bisnis. Kondisi disetiap alur diperiksa, kemudian alur yang memenuhi dijalankan.

Sumber: Object Management Group (2011)

### 2.5.2.2 Data

Data digunakan untuk menggambarkan dokumen, data, dan objek lain digunakan dan diperbarui pada sebuah alur proses. Penjelasan notasi terdapat pada Tabel 2.4.

**Tabel 2.4 Notasi Data**

Notasi	Nama	Deskripsi
	<i>Data Object</i>	Notasi untuk memberikan informasi mengenai data, dokumen atau objek lain yang berkaitan dengan aktivitas tertentu.
	<i>Data Object (Collection)</i>	Notasi untuk menggambarkan <i>data object</i> yang berjumlah lebih dari satu.
	<i>Data Input</i>	Notasi untuk menggambarkan data, dokumen, atau objek lain yang menjadi <i>input</i> dalam aktivitas.
	<i>Data Output</i>	Notasi untuk menggambarkan data, dokumen, atau objek lain yang menjadi <i>output</i> dalam aktivitas.
	<i>Data Store</i>	Notasi yang menggambarkan informasi mengenai sebuah lokasi penyimpanan dan pembaruan data.


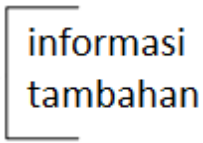
Sumber: Object Management Group (2011)

### 2.5.2.3 Artifacts

Berguna untuk menggambarkan informasi tambahan mengenai proses bisnis. terdiri dari *data object*, *groups*, dan *text annotations*. Setiap *artefacts* dapat dikaitkan dengan *flow elementer* (Weske, 2007). Penjelasan notasi terdapat pada Tabel 2.5.



**Tabel 2.5 Notasi *Artifacts***





Notasi	Nama	Deskripsi
	<i>Group</i>	Notasi untuk mengelompokkan notasi-notasi ke dalam sebuah kategori. Pada umumnya digunakan untuk kepentingan dokumentasi dan tidak mempengaruhi alur proses bisnis.
	<i>Text Annotation</i>	Notasi yang digunakan untuk memberikan informasi dalam bentuk teks untuk memudahkan pembaca BPMN.

Sumber: Object Management Group (2011)

#### 2.5.2.4 *Connecting Object*

*Connecting Object* berisi notasi untuk menghubungkan notasi flow object dengan notasi-notasi BPMN lainnya dikaitkan dengan *flow elementer* (Weske, 2007). Penjelasan notasi terdapat pada Tabel 2.6.

**Tabel 2.6 Notasi *Connecting Object***



Notasi	Nama	Deskripsi
	<i>Sequence Flow</i>	Berguna untuk menunjukkan urutan aliran dalam sebuah proses dan hanya boleh memiliki satu sumber <i>activity</i> dan satu target.
	<i>Message Flow</i>	Notasi yang digunakan untuk menunjukkan aliran yang dikirimkan dan diterima sebuah aktivitas dengan partisipan yang berbeda.
	<i>Associations</i>	Notasi untuk menghubungkan notasi-notasi <i>artifacts</i> yang berisi informasi dengan notasi BPMN lainnya.
	<i>Data Associations</i>	Notasi yang digunakan untuk menghubungkan aktivitas BPMN dengan notasi data.

Sumber: Object Management Group (2011)

#### 2.5.2.5 *Swimlaness*

*Swimlaness* ialah jenis notasi untuk mengelompokkan aktivitas berdasarkan unit bisnis yang menjalankan aktivitas-aktivitas tertentu (Object Management Group, 2011). Penjelasan notasi terdapat pada Tabel 2.7.

**Tabel 2.7 Notasi Swimlaness**

Notasi	Nama	Deskripsi
	<i>Pool</i>	Notasi yang merepresentasikan partisipan. Partisipan berupa suatu organisasi / perusahaan atau perwakilan peranan yang lebih umum (misalnya: produsen, konsumen).
	<i>Lane</i>	Notasi yang merupakan bagian dalam proses atau <i>pool</i> yang memiliki kesamaan panjang dengan proses, baik secara vertikal maupun horizontal. Notasi ini digunakan untuk mengatur dan mengelompokkan aktivitas dalam sebuah <i>pool</i> . <i>Lane</i> terkadang digunakan untuk merepresentasikan peran internal perusahaan

Sumber: Object Management Group (2011)

### 2.5.3 Pemodelan *Use case*

Pemodelan *use case* menggambarkan bagaimana pengguna menggunakan sistem dan yang dilakukan sistem untuk penggunaannya. Model *use case* berperan penting dalam penentuan persyaratan, khususnya dalam meningkatkan komunikasi antara pengembang perangkat lunak dengan pemangku kepentingan. Sehingga pertukaran informasi mengenai pemangku kepentingan yang terlibat dan terkena dampak akan lebih mudah dan tepat (Bittner & Spence, 2002).

Pemodelan *use case* dapat mengidentifikasi siapa saja calon pengguna sistem, mengetahui kebutuhan pemangku kepentingan dan pengguna sistem berdasarkan masalah yang dihadapi, serta untuk mengetahui kemungkinan solusi yang ditawarkan kepada pemangku kepentingan dan pengguna untuk memenuhi kebutuhannya.

#### 2.5.3.1 Identifikasi Tipe Pemangku Kepentingan

Pemangku kepentingan merupakan seseorang yang secara materi dipengaruhi oleh hasil sebuah sistem atau proyek yang memproduksi sistem. Sehingga diperlukannya identifikasi tipe pemangku kepentingan untuk mengetahui siapa saja pemangku kepentingan dari sebuah sistem. Pemangku Kepentingan tersebut

akan dikelompokkan berdasarkan karakteristik dan hubungan pemangku kepentingan dengan sistem (Bittner & Spence, 2002).

Berikut beberapa kategori pemangku kepentingan (Bittner & Spence, 2002) :

1. Pengguna (*Users*): memerankan peran sebagai aktor pada model *use case* dan berinteraksi langsung dengan sistem.
2. Sponsor (*Sponsors*): pengguna tidak langsung dari sistem yang berinvestasi kedalam pengembangan sistem.
3. Pengembang (*Developers*): pemangku kepentingan yang terlibat dalam kegiatan produksi dan pengembangan sistem.
4. Pihak berwenang (*Authorities*): pemangku kepentingan yang memiliki keahlian khusus dalam memberikan dukungan untuk pengembangan sistem dalam bidang regulasi internal atau eksternal sebuah perusahaan, teknologi, aturan perundangan.
5. Pelanggan (*Customers*): pemangku kepentingan yang akan membeli sistem setelah selesai dikembangkan.

**Tabel 2.8 Analisis Tipe Pemangku Kepentingan**

Tipe Pemangku Kepentingan	Deskripsi	Pemangku Kepentingan
Identifikasi tipe pemangku kepentingan	Mendeskripsikan tipe pemangku kepentingan	Menjelaskan aktor-aktor yang termasuk ke dalam tipe pemangku kepentingan

Sumber: Bittner & Spence (2002)

Bittner dan Spencer menjelaskan untuk menganalisis tipe pemangku kepentingan dengan menggunakan tabel. Pada Tabel 2.8 merupakan contoh tabel analisis tipe pemangku kepentingan.

### 2.5.3.2 Analisis Masalah

Analisis masalah merupakan proses untuk memahami masalah yang aktual sehingga dapat menjelaskan kebutuhan pengguna, serta proses untuk mengusulkan solusi untuk memenuhi kebutuhan pengguna (Leffingwell & Widrig, 2003).

Menurut Bittner dan Spence, cara terbaik untuk mendokumentasikan analisis masalah adalah dengan menyusun pernyataan masalah (*problem statement*). Pernyataan masalah akan menjelaskan masalah yang harus dipecahkan dan ringkasan solusi untuk mengatasi permasalahan. Pada penelitian ini pernyataan masalah diperoleh dari hasil wawancara dengan pemangku kepentingan, kemudian pernyataan masalah didokumentasikan pada tabel, berikut contoh tabel pernyataan masalah yang dijelaskan pada Tabel 2.9.

**Tabel 2.9 Kerangka Dokumentasi Pernyataan Masalah**

Masalah	Deskripsi Masalah
Mempengaruhi ( <i>Affects</i> )	Berisi pemangku kepentingan yang terkena pengaruh oleh masalah.
Dampak ( <i>The impact of which is</i> )	Berisi penjelasan dampak masalah terhadap pemangku kepentingan dan kegiatan bisnis.
Solusi ( <i>A successful solution would</i> )	Berisi beberapa solusi dan manfaat utama yang ditawarkan untuk menyelesaikan masalah.

Sumber: Bittner & Spence (2002)

### 2.5.3.3 Identifikasi Kebutuhan Pemangku Kepentingan dan Pengguna

Masalah yang telah di analisis perlu untuk dipecahkan dengan melibatkan pemenuhan kebutuhan dari pemangku kepentingan. Kebutuhan pemangku kepentingan dan pengguna atau secara singkat disebut kebutuhan pengguna merupakan sesuatu yang dapat mewakili masalah bisnis, personal atau operasional yang harus ditekankan dengan tujuan untuk membenarkan pertimbangan, pembelian, atau penggunaan sistem baru. Memperoleh informasi mengenai kebutuhan pengguna dapat dilakukan dengan beberapa teknik yaitu wawancara, *requirement workshop*, *brainstorming and idea reduction*, *storyboarding* (Leffingwell & Widrig, 2003).

Informasi kebutuhan pengguna yang diperoleh dapat membantu dalam memahami sejauh mana beberapa aspek yang berbeda dari sebuah masalah dapat mempengaruhi jenis pemangku kepentingan yang berbeda dan dapat menyediakan pemahaman yang lebih mengenai pernyataan masalah yang diberikan oleh pemangku kepentingan (Bittner & Spence, 2002).

### 2.5.3.4 Identifikasi Fitur

Fitur merupakan layanan dari sistem yang diperlukan untuk memberikan manfaat kepada pengguna dan membantu dalam memenuhi kebutuhan pemangku kepentingan. Fitur berguna untuk memberikan rangkuman mengenai kemampuan *functional* dari sistem yang dibangun (Bittner & Spence, 2002).

Bittner dan Spence menjelaskan penamaan fitur harus memiliki tingkat detail yang umum agar seluruh pemangku kepentingan dapat memahami kemampuan sistem. Fitur yang telah ditentukan dapat didokumentasikan dalam tabel seperti contoh pada tabel 2.10.

**Tabel 2.10 Contoh Tabel Fitur**

Kode Fitur	Deskripsi
Berisi kode fitur sebagai identitas	Berisi deskripsi dari kemampuan yang dapat dilakukan sistem.

Sumber: Bittner & Spence (2002)

### 2.5.3.5 Identifikasi Persyaratan Fungsional dan Persyaratan Nonfungsional

Pengelompokan persyaratan perangkat lunak secara umum dikategorikan menjadi persyaratan fungsional dan persyaratan nonfungsional. Dalam mendokumentasikan persyaratan perangkat lunak memiliki banyak cara, namun yang paling umum digunakan adalah dengan mendeskripsikan persyaratan tersebut dalam kalimat deklaratif (Bittner & Spence, 2002).

Persyaratan fungsional merupakan pernyataan yang menyatakan bagaimana sistem bereaksi terhadap *input* tertentu dan bagaimana sistem berperilaku dalam situasi tertentu. Sedangkan persyaratan nonfungsional merupakan kebutuhan yang secara tidak langsung terkait dengan layanan yang diberikan sistem kepada pengguna dan berhubungan dengan sifat sistem seperti *reliability*, waktu respon, dan keamanan dari sistem (Sommerville, 2011).

### 2.5.3.6 Pemodelan *Use case*

Model *use case* berguna untuk merepresentasikan bagaimana pengguna menggunakan sistem tersebut. Model *use case* merupakan kumpulan dari semua *use case*, aktor dan hubungannya. Aktor merupakan objek diluar sistem yang merepresentasi individu (manusia) atau oebjek lain yang berinteraksi dengan sistem. Sedangkan *use case* merepresentasikan hal-hal yang harus dilakukan sistem untuk aktornya. Aktor dihubungkan dengan *use case* sesuai dengan interaksi aktor dengan *use case* tertentu (Bittner & Spence, 2002).

Model *use case* tidak menjelaskan secara lengkap persyaratan sebuah sistem. Model *use case* memiliki spesifikasi *use case* yang berisi penjelasan mengenai bagaimana aktor dan sistem bekerja sama. Format dokumentasi spesifikasi *use case* ditunjukkan pada Tabel 2.11.

**Tabel 2.11 Format Dokumentasi Spesifikasi *Use case***

<b>Name</b>	Berisi nama dari <i>use case</i> yang berguna sebagai identitas dari <i>use case</i> .
<b>Brief Description</b>	Berisi penjelasan singkat mengenai peran dan tujuan dari <i>use case</i> .
<b>Actor</b>	Berisi aktor yang berinteraksi dengan <i>use case</i> .
<b>Pre-Condition</b>	Berisi deskripsi satu atau lebih kondisi yang harus dipenuhi oleh sistem terlebih dahulu sebelum <i>use case</i> dijalankan.
<b>Post-Condition</b>	Berisi deskripsi satu atau lebih kondisi yang dipenuhi oleh sistem ketika <i>use case</i> berakhir.
<b>Extension</b>	Berisi informasi penggunaan <i>use case</i> lain yang bersifat opsional.
<b>Basic Flow</b>	Berisi alur normal yang dilalui ketika <i>use case</i> berjalan.
<b>Alternative Flow</b>	Berisi alur opsional yang akan dilalui ketika <i>use case</i> tidak berjalan sesuai dengan alur normal ( <i>basic flow</i> ).



<b>Sub Flow</b>	Berisi beberapa kelompok alur untuk menyederhanakan alur <i>use case</i> yang terlalu detail.
-----------------	---

Sumber: Diadaptasi dari Bittner & Spence (2002)

## 2.6 Tahap Perancangan Sistem

Tahap perancangan sistem atau *system and software design* merupakan tahap kedua dalam metode *waterfall*. Perancangan sistem merupakan proses inovatif yang menerjemahkan persyaratan atau kebutuhan menjadi *blueprint* untuk membangun perangkat lunak. *Blueprint* menggambarkan perangkat lunak melalui desain abstraksi tingkat tinggi, yaitu dengan menggambarkan fungsional dan perilaku dari sistem. Kemudian saat iterasi desain selanjutnya, fungsional dan perilaku tersebut direpresentasikan menjadi desain abstraksi tingkat rendah (Pressman, 2010).

Menurut Davis pada (Pressman, 2010), menyarankan prinsip untuk perancangan perangkat lunak yaitu :

- 1) Perancangan yang baik harus mempertimbangkan pendekatan alternatif, menilai setiap masalah persyaratan, memahami sumber daya yang tersedia.
- 2) Desain harus bersifat *traceable*, agar dapat melacak bagaimana persyaratan telah dipenuhi oleh model desain.
- 3) Desain dibangun menggunakan satu set pola desain.
- 4) Desain harus meminimalkan jarak intelektual antara perangkat lunak dan masalah yang ada. Sehingga perangkat lunak sebisa mungkin dapat meniru struktur domain masalah.
- 5) Desain harus menunjukkan keseragaman dan integrasi.
- 6) Desain harus dibuat untuk mengakomodasi perubahan.
- 7) Desain harus dapat mengakomodasikan keadaan yang tidak biasa (error) atau permasalahan, dan harus dapat mengatasi permasalahan tersebut.
- 8) Desain tidak sama dengan *coding*, sehingga tingkat abstraksi dari desain harus lebih tinggi dari *source code*.
- 9) Desain harus dinilai kualitasnya setelah dibuat.
- 10) Desain harus ditinjau untuk meminimalkan kesalahan konseptual.

Pada penelitian ini perancangan sistem dilakukan dengan memodelkan rancangan menggunakan bahasa pemodelan *Unified Modelling Language*.

### 2.6.1 Unified Modelling Language

*Unified Modelling Language* merupakan bahasa untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan artefak sistem perangkat lunak, pemodelan bisnis, dan sistem non-perangkat lunak lainnya

(Object Management Group, 2005). Diagram-diagram pada UML terdiri dari *Use case Diagram*, *Sequence Diagram*, serta *Class Diagram*


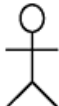
Sedangkan menurut Pressman (2010), UML adalah bahasa standar yang digunakan dalam penulisan *blueprint* perangkat lunak. UML digunakan untuk memvisualisasikan, melakukan spesifikasi, konstruksi, dan mendokumentasikan artifak dari sistem perangkat lunak. Peneliti memanfaatkan beberapa diagram dalam penelitian ini terdiri dari *use case diagram*, *class diagram* (varian dari *behavior diagram*), dan *sequence diagram* (varian dari *interaction diagram*, yang juga merupakan varian dari *behavior diagram*).


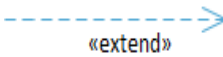
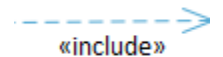

### 2.6.1.1 Use case Diagram

*Use case Diagram* menggambarkan perilaku sistem, sub sistem, kelas atau komponen yang tampak pada pengguna dan menggambarkan interaksi dengan aktor sebagai urutan pesan antara sistem dan aktor (Booch, et al., 2005). *Use case* digunakan untuk mengetahui fungsi dan fitur perangkat lunak dari persepektif pengguna (Pressman, 2010).

*Use case diagram* menampilkan hubungan di antara aktor dengan *use case* (Object Management Group, 2005). Hubungan yang tergambar di dalam *use case diagram* adalah hubungan asosiasi antara aktor dengan satu atau lebih *use case*, hubungan generalisasi antar aktor, dan hubungan generalisasi, *extends*, dan *includes* antar *use case*. Sekumpulan *use case* secara opsional dapat diletakkan di dalam sebuah notasi berbentuk persegi yang merepresentasikan *boundary* sebuah sistem.

Tabel 2.12 Notasi Use case

Notasi	Nama	Deskripsi
 A Use Case	Use case	Notasi yang menggambarkan spesifikasi tingkah laku sebuah entitas terhadap interaksi yang terjadi di luar. Mendeskripsikan bagaimana aktor menggunakan sistem untuk mencapai tujuan tertentu dan apa yang sistem lakukan untuk aktor agar mendapatkan tujuannya.
 An Actor	Actor	Penggambaran pengguna di luar sistem, sistem atau sub sistem yang memiliki peran terhadap sistem. Setiap aktor berpartisipasi terhadap satu atau lebih <i>use case</i> di dalam sistem. Seorang aktor bisa berupa manusia, sistem komputer atau proses eksekusi yang lainnya.

Notasi	Nama	Deskripsi
	<i>Association</i>	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> .
	<i>Extend</i>	Tingkah laku yang hanya dilakukan jika ada kondisi tertentu.
	<i>Include</i>	Tingkah laku yang wajib dipenuhi terlebih dahulu agar <i>use case</i> dapat berjalan. <i>Use case</i> tersebut merupakan bagian dari <i>use case</i> lainnya.
	<i>Generalization</i>	Hubungan antara <i>use case</i> yang umum dan lebih spesifik.


Sumber: Grady Booch, James Rumbaugh, Ivar Jacobson (2005)


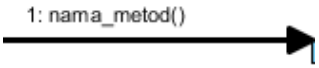
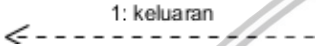
Sedangkan *use case specification* digunakan untuk menggambarkan urutan kegiatan yang dilakukan oleh aktor dan sistem untuk mencapai suatu tujuan tertentu. *use case specification* terdiri dari *Use case Name*, *Brief Description* (deskripsi singkat), *Basic Flow* (aliran normal), *Alternate Flow* (aliran alternatif), *Pre Condition* dan *Post Condition* (Pressman, 2010). Notasi yang digunakan pada diagram *use case* dapat dilihat pada Tabel 2.12.

#### 2.6.1.2 Sequence Diagram

Diagram *Sequence* merupakan diagram interaksi yang menekankan pada saat permintaan pesan yang menunjukkan serangkaian peranan dan pesan yang dikirim dan diterima oleh bagian yang ada pada peranan tersebut. *Sequence diagram* digunakan untuk menggambarkan dynamic view dari sistem (Booch, et al., 2005). Notasi yang digunakan pada *sequence diagram* dapat dilihat pada Tabel 2.14. dan 2.15.



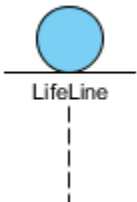
Tabel 2.13 Notasi *Sequence Diagram*

Notasi	Nama	Deskripsi
	<i>Lifeline</i>	Menunjukkan keberadaan objek selama periode waktu.

Notasi	Nama	Deskripsi
	Activation bar	Menunjukkan periode waktu ketika aktif dalam interaksi.
	Pesan tipe <i>call</i> ( <i>synchronous</i> )	Mendefinisikan pengiriman pesan. Arah panah mengarah pada objek yang memiliki fungsi, karena ini memanggil sebuah fungsi maka fungsi yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.
	Pesan tipe <i>return</i>	Mendefinisikan hasil dari pengiriman pesan. Arah panah mengarah pada objek yang menerima kembalian.

Sumber: Grady Booch, James Rumbaugh, Ivar Jacobson (2005)

Tabel 2.14 Notasi *Sequence Diagram* (lanjutan)

Notasi	Nama	Deskripsi
	Lifeline <<Boundary>>	Menunjukkan keberadaan suatu kelas <i>interface</i> yang menghubungkan pengguna dengan sistem selama periode waktu.
	Lifeline <<Control>>	Menunjukkan keberadaan suatu <i>control</i> yang dilakukan oleh sebuah kegiatan selama periode waktu.
	Lifeline <<Entity>>	Menunjukkan keberadaan suatu <i>entity</i> untuk keperluan penyimpanan data selama periode waktu.

Sumber: Visual Paradigm

### 2.6.1.3 Class Diagram

*Class diagram* digunakan untuk memodelkan tampilan statis perancangan dari sebuah sistem. *Class diagram* memiliki beberapa tingkat detail yang berbeda-beda. Beberapa tingkat detail yang digunakan dalam beberapa tahap pengembangan perangkat lunak di antaranya pada tahap persyaratan, analisis dan perancangan.

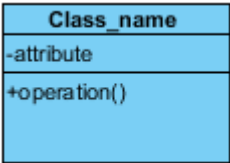
*Class diagram* pada tahap analisis dapat dihasilkan berdasarkan analisis yang dilakukan pada *use case diagram*. Setiap *use case diagram* akan diidentifikasi untuk menghasilkan objek-objek yang ada. Kemudian setiap objek tersebut dapat digabungkan untuk menjadi diagram kelas analisis (IBM, 2004). Diagram kelas analisis dapat dijadikan panduan untuk membuat diagram kelas perancangan dan perancangan basis data. Kemudian untuk mendapatkan *class diagram* pada tahap perancangan dapat dilakukan dengan pemetaan berdasarkan diagram kelas analisis, terdiri dari :

1. Satu kelas analisis menjadi satu kelas desain.
2. Satu kelas analisis bisa menjadi bagian dari kelas desain.
3. Sekelompok kelas analisis mewarisi dari kelas yang sama pada kelas desain.





*Class diagram* secara umum membuat tiga hal yaitu kelas, *interfaces*, dan hubungan dependensi, generalisasi, serta asosiasi (Booch, et al., 2005). Tabel 2.16 merupakan notasi dari *class diagram*.

- 1) Kelas, adalah deskripsi dari sekumpulan objek yang memiliki atribut, operasi, hubungan, dan semantic yang sama.
- 2) *Interfaces* dalam UML digunakan untuk memodelkan lapisan dalam suatu sistem yang merupakan kumpulan dari operasi yang digunakan untuk menentukan sebuah layanan dari kelas atau komponen.
- 3) Dependensi, merupakan hubungan yang menyatakan bahwa sebuah kelas menggunakan informasi dan layanan dari kelas lain, tetapi belum tentu sebaliknya.
- 4) Generalisasi merupakan hubungan antara sebuah hal yang umum (*superclass* atau *parent*) dan jenis hal yang lebih spesifik (*subclass* atau *child*).
- 5) Asosiasi merupakan hubungan struktural yang menentukan bahwa objek dari satu kelas terhubung dengan objek kelas lain (Booch, et al., 2005).

**Tabel 2.15 Notasi Class Diagram**

Notasi	Nama	Deskripsi
	<i>Class</i>	<p>Notasi yang menggambarkan struktur kelas pada suatu sistem</p> <p><i>Attribute</i> mendefinisikan kumpulan data yang dimiliki oleh suatu objek,</p>

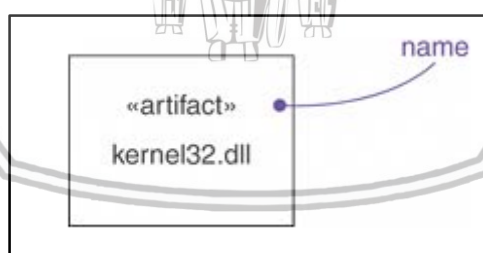


Notasi	Nama	Deskripsi
		<i>Operation</i> mendefinisikan fungsi yang bertindak pada data objek (atribut).
	<i>Association</i>	Mengindikasikan objek dari satu kelas berhubungan dengan kelas lain.
	<i>Dependency</i>	Ketergantungan antar kelas.
	<i>Direct association</i>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain.
	Generalisasi	Mengindikasikan relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain.

Sumber: Grady Booch, James Rumbaugh, Ivar Jacobson (2005)

#### 2.6.1.4 Artefak Sistem

Dalam UML, artefak sistem adalah benda fisik pada tingkat platform implementasi. Selain benda fisik, artefak sistem dapat digunakan untuk memodelkan hal-hal yang berpartisipasi dalam sistem, seperti tabel, file, dan dokumen. UML memberikan representasi grafis dari artefak, seperti yang ditunjukkan pada Gambar 2.4. Notasi tersebut memungkinkan untuk memvisualisasikan artefak sistem dengan menggunakan *stereotypes*. Notasi ini akan mewakili jenis artefak tertentu (Booch, et al., 2005).

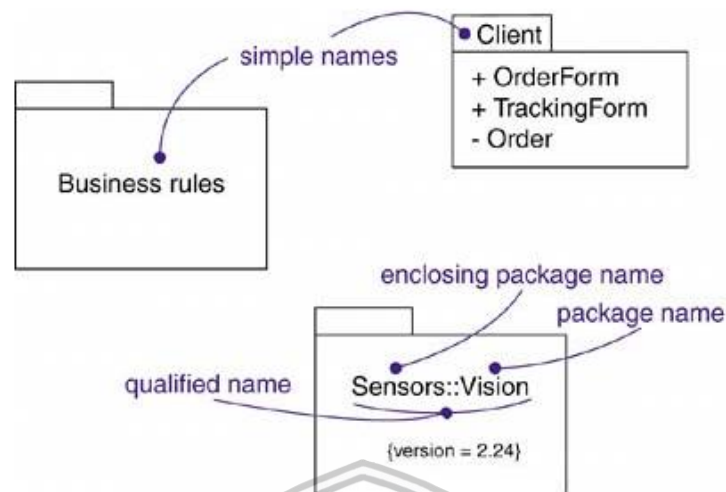


**Gambar 2.4 Pemodelan Artefak Sistem**

Sumber: Grady Booch, James Rumbaugh, Ivar Jacobson (2005)

#### 2.6.1.5 Package

*Package* adalah mekanisme tujuan umum untuk mengatur hierarki dari suatu pemodelan. Package dinotasikan menjadi sebuah folder yang memiliki nama. *Package* dapat mengelompokkan suatu pemodelan ataupun artefak sistem sesuai dengan tujuannya masing-masing (Booch, et al., 2005). Contoh dari pemodelan *package* dapat dilihat pada Gambar 2.5.



**Gambar 2.5 Pemodelan *Package***

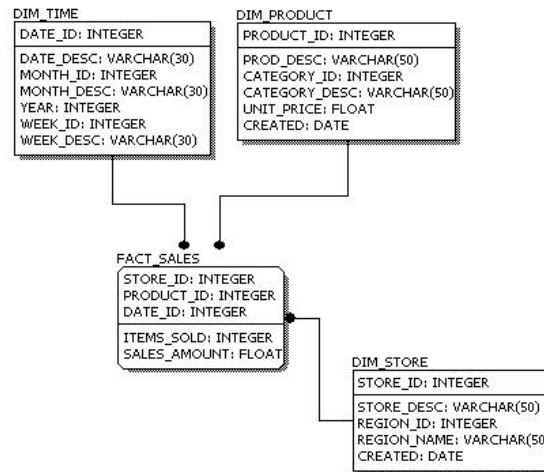
Sumber: Grady Booch, James Rumbaugh, Ivar Jacobson (2005)

### 2.6.2 *Physical Data Model*

*Physical Data Model* merupakan sebuah model spesifik untuk basis data yang merepresentasikan objek data relasional seperti tabel, kolom, *primary key* dan *foreign key* beserta relasinya. *Physical Data Model* dapat digunakan untuk menghasilkan pernyataan dalam *Data Definition Language* (DDL) yang kemudian diterapkan ke dalam *server* basis data (IBM, 2018).

Pemodelan data dalam PDM dapat dilakukan berdasarkan hasil pemodelan kelas dalam *class diagram*. Contoh PDM terdapat dalam Gambar 2.6. Booch, et al. (1998), menjelaskan, pemodelan PDM dapat dilakukan dengan 3 strategi, di antaranya yaitu:

1. Mendefinisikan masing-masing kelas pada *class diagram* menjadi tabel pada PDM. Hal ini merupakan strategi yang paling sederhana, namun akan mendatangkan kesulitan ketika rancangan *class diagram* mengalami perubahan, misal, terdapat *subclass* yang ditambahkan.
2. Mengumpulkan *subclass* dengan *superclass* pada tingkat hirarki yang sama, kemudian definisikan masing-masing kelas menjadi tabel pada PDM. Kekurangan strategi ini adalah akan menyebabkan kapasitas penyimpanan cepat berkurang.
3. Memisahkan *subclass* dengan *superclass* dan mendefinisikannya menjadi tabel pada PDM. Kekurangan dari strategi ini adalah akan membutuhkan mekanisme *join table* untuk proses pengambilan data melalui basis data.



**Gambar 2.6 Contoh Physical Data Model**

Sumber: 1keydata (2009)

## 2.7 Tahap Implementasi Sistem

Tahap implementasi sistem merupakan tahap ketiga dalam metode *waterfall*. Implementasi Sistem merupakan fase untuk merealisasikan perancangan sistem sebagai satu set program atau unit program, dimana desain / rancangan akan diterjemahkan ke dalam bentuk bahasa pemrograman tertentu (Sommerville, 2011).

### 2.7.1 Object Oriented Programming (OOP)

Pemrograman berorientasi obyek merupakan pembangunan perangkat lunak yang memandang perangkat lunak sebagai kumpulan dari beberapa obyek yang berisi data dan fungsi. Metode OOP banyak digunakan karena memiliki keuntungan yaitu (Shalahuddin & S, 2015) :

- 1) *Reusable* Karena obyek yang digunakan dalam suatu masalah dapat dipakai ulang.
- 2) Meningkatkan kecepatan pengembangan karena sistem telah dianalisis dan dirancang dengan baik dan benar sehingga mengurangi kesalahan pada saat implementasi kode.
- 3) Kemudahan pemeliharaan karena menggunakan obyek yang dapat diubah-ubah jika terdapat perubahan.
- 4) Dengan pewarisan, memungkinkan dapat peningkatan konsistensi.
- 5) Meningkatkan kualitas perangkat lunak karena pengembangan didasari dengan kejadian pada dunia nyata.

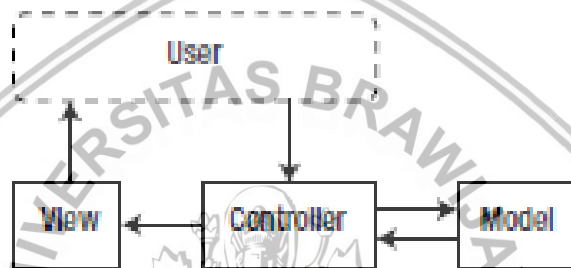
### 2.7.2 Model-View-Controller (MVC)

*Model*, *View* dan *Controller* (MVC) merupakan salah satu pola desain perangkat lunak yang dibangun berdasarkan kesinambungan antara 3 jenis komponen utama dan pada pemrograman seperti PHP yang sering digunakan

sebagai fokus utama dalam pengembangan berorientasi objek (Pitt, 2012). Gambar 2.7 merupakan struktur dari MVC.

MVC terdiri dari 3 bagian (Pitt, 2012), yaitu:

- 1) *Model*: bagian dimana seluruh logika bisnis diletakkan dan bagaimana aplikasi mengakses dan menyimpan data ke *database* untuk memenuhi kebutuhan alur bisnis tersebut.
- 2) *View*: bagian yang berkaitan dengan antarmuka pengguna berada. Seluruh hal yang dilihat pengguna berasal dari *view*.
- 3) *Controller*: bagian untuk menghubungkan dan mengatur alur yang terjadi antara *model* dan *view*. *Controller* memberi sekat agar proses yang ada pada model tidak terlihat oleh pengguna di *view*.



Gambar 2.7 Struktur MVC

Sumber: Chris Pitt (2012)

## 2.8 Tahap Pengujian Sistem

Tahap pengujian sistem merupakan tahap keempat dalam metode *waterfall*. Pengujian perangkat lunak dilakukan untuk menunjukkan bahwa perangkat lunak yang dikembangkan dapat berjalan sesuai dengan tujuan dan untuk mengetahui kekurangan (*defects*). Pengujian perangkat lunak dilakukan dengan data buatan. Proses pengujian memiliki dua tujuan yaitu (Sommerville, 2011) :

- 1) Untuk menunjukkan kepada pelanggan dan pengembang bahwa perangkat lunak memenuhi persyaratan.
- 2) Untuk menemukan situasi jika perangkat lunak tidak berjalan sesuai yang diharapkan atau tidak sesuai dengan spesifikasinya.

Pada penelitian ini, peneliti mengguna dua metode pengujian, yaitu pengujian validasi, dan juga pengujian komparabilitas peramban.

### 2.8.1 Black-Box Testing

*Black-box testing* merupakan pengujian perangkat lunak yang berfokus pada persyaratan fungsional perangkat lunak. Pengujian ini dilakukan untuk mengetahui kesalahan / *error* yang terdapat perangkat lunak. Kesalahan / *error* yang dimaksud antara lain (Pressman, 2010):

1. Kesalahan atau hilangnya fungsi.
2. Kesalahan antarmuka.
3. Kesalahan pada struktur data atau pada akses basis data eksternal.
4. Kesalahan pada kinerja atau performa perangkat lunak
5. Kesalahan inisialisasi atau terminisasi di dalam perangkat lunak.

*Black-box testing* memiliki beberapa keunggulan, yaitu:

1. Pengujian dilakukan berdasar sudut pandang pengguna untuk membantu dalam mengungkap ketidaksesuaian dalam spesifikasi perangkat lunak.
2. Penguji tidak perlu mengetahui bahasa pemrograman, atau bagaimana mengimplementasikan perangkat lunak.
3. Pengujian dapat dilakukan oleh badan independen dari pengembang, dan memungkinkan memiliki perspektif objektif.
4. Test case dapat langsung dirancang setelah spesifikasinya lengkap.

#### **2.8.1.1 Pengujian Validasi**

Pengujian validasi merupakan pengujian yang berfokus pada hasil (output) dari sistem yang terlihat oleh pengguna. Pengujian ini dilakukan untuk mengetahui apakah sistem yang dibangun berjalan sesuai dengan kebutuhan yang telah didefinisikan. Pengujian validasi dilakukan dengan cara mendemonstrasikan perangkat lunak yang dikembangkan (Pressman, 2010).

Pengujian validasi perangkat lunak dapat dilakukan dengan menggunakan *use case* sebagai panduan pengujian. *Use case* menyediakan skenario yang dapat mendeteksi kemungkinan deviasi atau kesalahan pada perangkat lunak. Setelah pengujian selesai dilakukan, akan didapatkan dua hasil, yaitu (Pressman, 2010) :

1. Fungsi atau performa sistem telah sesuai dengan spesifikasi dan diterima oleh pelanggan.
2. Deviasi dan kekurangan perangkat lunak yang dikembangkan terlihat dan kemudian akan dicatat.

#### **2.8.1.2 Pengujian Kompatibilitas Peramban**

Pengujian kapabilitas peramban merupakan salah satu pengujian yang dilakukan terhadap antarmuka pengguna pada aplikasi situs web. Pengujian ini berguna untuk mengetahui apakah aplikasi memiliki masalah ketika berjalan pada komputer, tampilan perangkat, sistem operasi, *browser* dan kecepatan koneksi jaringan yang berbeda. Pengujian dilakukan dengan melakukan konfigurasi komputasi perangkat lunak pada perangkat pengguna (klien). Konfigurasi dapat meliputi penggunaan sistem operasi yang berbeda, perangkat tampilan berbeda, aplikasi peramban yang berbeda atau kecepatan jaringan yang berbeda. Hal ini dilakukan untuk mengetahui kesalahan pada eksekusi aplikasi pada beberapa konfigurasi yang berbeda (Pressman, 2010).



## BAB 3 METODOLOGI

Penelitian ini merupakan penelitian implementatif. Sehingga luaran produk yang dihasilkan adalah sistem informasi. Pendekatan implementatif yang digunakan adalah analisis dan perancangan sistem menggunakan pendekatan berorientasi objek. Gambar 3.1 merupakan alur dan langkah-langkah dalam penelitian ini.



**Gambar 3.1 Diagram Alur Penelitian**

### 3.1 Wawancara

Pada tahap ini, peneliti melakukan pengumpulan data dengan menggunakan teknik wawancara. Wawancara awal dilakukan kepada Kepala Sekolah Dasar Negeri Dinoyo 1 untuk mengetahui permasalahan yang ada pada sekolah yaitu pada proses pendaftaran siswa baru dan pendataan ulang data pokok siswa. Kemudian peneliti melakukan wawancara terhadap Kepala Tata Usaha SDN Dinoyo 1 untuk mengetahui alur proses pendaftaran siswa baru dan juga proses pendataan ulang data pokok siswa saat ini dan permasalahan yang ada pada

proses tersebut. Hasil dari wawancara akan digunakan untuk menganalisis dan menentukan kebutuhan dari sistem yang akan dikembangkan.

### 3.2 Studi Pustaka

Tahap kedua, peneliti melakukan pencarian, pengumpulan dan mempelajari referensi dari jurnal penelitian, buku, *e-book*, dan internet untuk mendapatkan penjelasan teori-teori yang dijadikan dasar untuk mendukung penelitian. Studi literature tersebut meliputi :

1. Penelitian sejenis yang pernah dilakukan
2. Sistem Informasi
3. Sistem Informasi Manajemen Akademik
4. *Bussiness Process Model and Notation*
5. *Software Development Life Cycle*
6. *Waterfall Model*
7. *Unified Modelling Language (UML)*
8. *Object Oriented Programming*
9. Pengujian Perangkat Lunak

### 3.3 Analisis Persyaratan

Tahap ketiga merupakan penerapan dari tahap *Requirements Definition* pada metode *waterfall*. Data yang didapat dari tahap Wawancara pada Lampiran A digunakan sebagai dasar analisis persyaratan sistem yang akan dikembangkan. Data yang didapat akan dirubah menjadi model proses bisnis saat ini. Kemudian dilakukan pemodelan proses bisnis setelah adanya sistem untuk menggambarkan solusi yang ditawarkan. Hasil pemodelan proses bisnis *to-be* akan menjadi sumber informasi pada analisis kebutuhan dari setiap *stakeholder* dengan menggunakan metode Pemodelan *Use case*. Pada Pemodelan *Use case* dilakukan identifikasi tipe pemangku kepentingan, analisis masalah, identifikasi kebutuhan pemangku kepentingan, identifikasi fitur, identifikasi persyaratan fungsional dan non fungsional.

### 3.4 Perancangan Sistem

Tahap perancangan merupakan penerapan dari tahap *System and Software Design* pada metode *waterfall*. Perancangan dilakukan dengan memodelkan sistem ke dalam notasi-notasi pada *Unified Modeling Language (UML)* berdasarkan hasil tahap Analisis Kebutuhan Sistem. Perancangan sistem berisi pemodelan objek ke dalam *class diagram*, hasil pemodelan interaksi objek ke dalam *sequence diagram*, perancangan basis data didokumentasikan ke dalam PDM, perancangan algoritma ke dalam *pseudocode*, dan perancangan antarmuka pengguna.

### 3.5 Implementasi Sistem

Tahap implementasi merupakan penerapan dari tahap *Implementation and Unit Testing* pada metode *waterfall*. Pada tahap ini dilakukan pembuatan sistem informasi manajemen akademik berdasarkan hasil dari tahap perancangan sistem. Implementasi dilakukan dengan menggunakan bahasa pemrograman *PHP*, *HTML*, *CSS*, dan *Javascript*, dengan kerangka kerja *Codeigniter*. Hasil dari tahap ini adalah Sistem Informasi Manajemen Akademik berbasis web yang menyediakan fitur sesuai dengan yang telah diidentifikasi pada tahap Analisis Kebutuhan dan Perancangan Sistem.

### 3.6 Pengujian Sistem

Tahap pengujian merupakan penerapan dari tahap *System Testing* pada metode *waterfall*. Pengujian diharapkan dapat memberikan umpan balik dari pengguna sistem informasi yang dapat dijadikan panduan perbaikan. Pengujian dilakukan dengan menggunakan metode pengujian validasi untuk mengetahui apakah sistem informasi yang dikembangkan telah memenuhi kebutuhan pengguna. Tahap pengujian validasi dilakukan dengan mendemonstrasikan dan menguji beberapa fungsi pada sistem informasi kepada calon pengguna yaitu petugas tata usaha SDN 1 Dinoyo. Pengujian juga dilakukan peneliti untuk menguji beberapa fungsi pada sistem informasi. Kemudian dilakukan pengujian kompatibilitas untuk mengetahui apakah aplikasi memiliki masalah ketika berjalan pada komputer, tampilan perangkat, sistem operasi, *browser* dan kecepatan koneksi jaringan yang berbeda.

### 3.7 Kesimpulan Dan Saran

Kesimpulan dilakukan pada tahap akhir penelitian ini. Kesimpulan bersisi jawaban dari masalah yang telah dirumuskan. Selain itu pada akhir penulisan terdapat saran yang dapat dijadikan pertimbangan untuk dilakukan pada penelitian selanjutnya.

## BAB 4 ANALISIS PERSYARATAN

Bab ini membahas analisis persyaratan yang merupakan penerapan dari metode *waterfall* pada tahap *requirements definition*. Analisis persyaratan diperoleh berdasarkan analisis pada hasil wawancara yang terdapat pada Lampiran B. Hasil analisis persyaratan terdiri dari model proses bisnis, tipe pemangku kepentingan, pernyataan masalah, kebutuhan pengguna, pengguna sistem informasi, fitur sistem informasi, persyaratan fungsional, non-fungsional, diagram *use case*, dan *use case spesifikasi*.

### 4.1 Aturan Penomoran

Aturan penomoran digunakan sebagai identitas untuk memudahkan kodifikasi dan *traceability*. Berikut aturan penomoran yang digunakan, serta realisasi aturan pada Tabel 4.1.



Kode Representasi Artefak Sistem :

1. AA = Aktivitas Proses Bisnis Saat ini (*as-is*)
2. AT = Aktivitas Proses Bisnis Usulan (*to-be*)
3. IM = Indetifikasi Masalah
4. N = Kebutuhan (*needs*)
5. FEAT = Fitur (*Feature*)
6. FR = Kebutuhan Fungsional (*Functional Requirements*)
7. NFR = Kebutuhan Non-Fungsional (*Non-Functional Requirements*)
8. UC = *Use case*

**Tabel 4.1 Realisasi Aturan Penomoran**

Artefak	Aturan Penomoran	Contoh Realisasi
Aktivitas Proses Bisnis Saat ini ( <i>as-is</i> )	AA-SIAS-XX	AA-SIAS-01
Aktivitas Proses Bisnis Usulan ( <i>to-be</i> )	AT-SIAS-XX	AT-SIAS-01
Indetifikasi Masalah	IM-SIAS-XX	IM-SIAS-01
Kebutuhan ( <i>needs</i> )	N-SIAS-XX	N-SIAS-01
Fitur ( <i>Feature</i> )	FEAT-SIAS-XX	FEAT-SIAS-01

Kebutuhan Fungsional ( <i>Functional Requirements</i> )	FR-SIAS-XX	FR-SIAS-01
Kebutuhan non-fungsional ( <i>Non-Functional Requirements</i> )	NFR-SIAS-XX	NFR-SIAS-01
<i>Use case</i>	UC-SIAS-XX	NFR-SIAS-01

## 4.2 Pemodelan Proses Bisnis

Pemodelan proses bisnis dilakukan untuk memodelkan serangkaian aktivitas pada SDN Dinoyo 1 dengan notasi yang lebih mudah dipahami, untuk memenuhi kebutuhan pengguna bisnis (internal maupun eksternal). Informasi mengenai proses bisnis yang akan dimodelkan didapatkan dengan cara melakukan wawancara dengan Kepala Tata Usaha SDN Dinoyo 1 mengenai alur proses bisnis pendaftaran siswa baru dan pendataan ulang data pokok siswa.

Dalam penelitian ini, pemodelan proses bisnis akan menghasilkan beberapa diagram proses bisnis yang akan terbagi menjadi 2 kelompok, yaitu proses bisnis saat ini berjalan (*as-is*) dan proses bisnis usulan (*to-be*). Tabel 4.2 menunjukkan hasil identifikasi aktor yang terlibat pada proses bisnis pendaftaran siswa baru dan pendataan ulang data pokok siswa. Hasil identifikasi aktor proses bisnis didapatkan dengan cara melakukan wawancara dengan Kepala Tata usaha SDN 1 Dinoyo.

**Tabel 4.2 Identifikasi Aktor pada Proses Bisnis**

Nama Aktor	Deskripsi
Petugas Tata Usaha	Aktor Petugas Tata Usaha diperankan oleh pegawai Tata Usaha SDN Dinoyo 1 yang bertugas untuk mengelola data pokok siswa.
Guru	Aktor Guru diperankan oleh Guru yang bertindak sebagai penguji untuk menerima berkas, menyeleksi berkas dan menetapkan siswa yang diterima pada proses bisnis pendaftaran siswa baru.
Orang Tua Calon Siswa	Aktor Orang Tua Calon Siswa diperankan oleh orang tua calon siswa yang melakukan pendaftaran pada proses bisnis pendaftaran siswa baru.
Orang Tua Siswa	Aktor Orang Tua Siswa diperankan oleh orang tua siswa yang melakukan pembaharuan data pokok pada proses bisnis pendataan ulang data pokok siswa.

### 4.2.1 Analisis Proses Bisnis As-Is

Analisis proses bisnis *as-is* dilakukan dengan melakukan wawancara dengan Kepala Tata Usaha SDN Dinoyo 1 mengenai alur aktivitas bisnis proses pendaftaran siswa baru dan pelaporan evaluasi hasil belajar yang saat ini berjalan. Kemudian



hasil analisis dimodelkan dengan notasi BPMN untuk membantu pemangku kepentingan memahami hasil analisis proses bisnis yang saat ini berjalan. Analisis dan pemodelan proses bisnis *as-is* yang dihasilkan dapat membantu peneliti untuk mengidentifikasi kemungkinan perbaikan atau peningkatan terhadap proses bisnis melalui solusi yang ditawarkan dalam penelitian ini. Pada Tabel 4.3 menjelaskan hasil identifikasi proses bisnis yang saat ini berjalan (*as-is*). Informasi mengenai identifikasi proses bisnis *as-is* diperoleh dari hasil wawancara. Hasil identifikasi terdiri dari nama proses bisnis, aktor yang terlibat, dan juga deskripsi singkat mengenai proses bisnis yang saat ini berjalan.

**Tabel 4.3 Hasil Identifikasi Proses Bisnis As-Is**

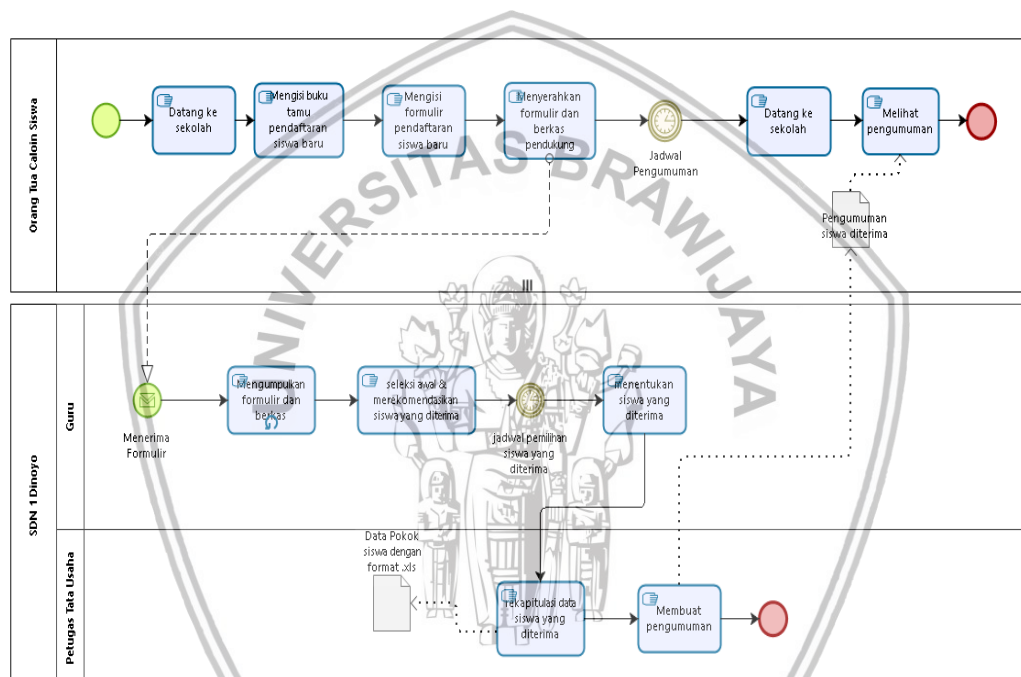
<b>Nama Proses Bisnis</b>	<b>Aktor yang Terlibat</b>	<b>Deskripsi</b>
Pendaftaran Siswa Baru	Orang Tua Calon Siswa, Guru, dan Petugas Tata Usaha.	Rangkaian aktivitas saat ini pendaftaran siswa baru yang dilakukan oleh Guru, Petugas Tata Usaha, dan Orang Tua Calon Siswa
Pendataan Ulang Data Pokok Siswa	Orang Tua Siswa, Guru, Siswa, dan Petugas Tata Usaha.	Rangkaian aktivitas saat ini pendataan ulang data pokok siswa yang dilakukan oleh Guru, Siswa, Petugas Tata Usaha, dan Orang Tua Siswa

#### **4.2.1.1 Proses Bisnis As-Is Pendaftaran Siswa Baru**

Dalam Gambar 4.1 merupakan rangkaian aktivitas pendaftaran siswa baru yang dilakukan oleh Guru, Petugas Tata Usaha SDN Dinoyo 1 dan Orang Tua Calon Siswa. Pada proses pendaftaran siswa baru, Orang Tua Calon Siswa dapat melakukan pendaftaran secara bersamaan dengan Orang Tua Calon Siswa yang lain tanpa harus saling menunggu proses pendaftaran selesai, sehingga kegiatan pada Orang Tua Calon Siswa dapat dilakukan secara parallel. Untuk melambangkan kegiatan tersebut maka digunakan notasi *multiple instance* bertipe parallel pada *pool* Orang Tua Calon Siswa. Uraian proses bisnis pendaftaran siswa baru saat ini yang terdapat pada Gambar 4.1 adalah sebagai berikut:

1. Proses bisnis pendaftaran siswa baru dimulai dengan aktivitas orang tua calon siswa datang ke sekolah untuk mendapatkan informasi mengenai pendaftaran siswa baru.
2. Orang tua calon siswa mengisi buku tamu pendaftaran di sekolah.
3. Orang tua calon siswa mengambil formulir pendaftaran siswa baru.
4. Orang tua calon siswa mengisi formulir pendaftaran dan menyiapkan berkas pendukung.
5. Orang tua calon siswa menyerahkan formulir pendaftaran dan berkas pendukung.

6. Guru menerima formulir pendaftaran dan berkas pendukung dari calon orang tua siswa. Kemudian guru mengumpulkan semua formulir dan berkas pendaftaran yang telah diisi.
7. Guru melakukan seleksi awal dan mencatat siswa yang direkomendasikan.
8. Guru menunggu jadwal pemilihan siswa yang diterima, kemudian menentukan siswa yang diterima.
9. Petugas Tata Usaha melakukan rekapitulasi semua data siswa yang diterima. Hasil dari rekapitulasi akan dijadikan data pokok siswa.
10. Petugas Tata Usaha membuat pengumuman daftar siswa yang diterima, dengan memasang pada papan pengumuman pada tanggal yang ditentukan.
11. Orang tua calon siswa datang ke sekolah untuk melihat pengumuman penerimaan siswa baru.



**Gambar 4.1 Proses Bisnis As-Is Pendaftaran Siswa Baru**

Tabel 4.4 menjelaskan mengenai daftar aktivitas pada proses bisnis pendaftaran siswa baru. Daftar aktivitas terdiri dari nama aktor, kode aktivitas, nama aktivitas, dan deskripsi singkat setiap aktivitas.

**Tabel 4.4 Daftar Aktivitas Proses Bisnis As-Is Pendaftaran Siswa Baru**

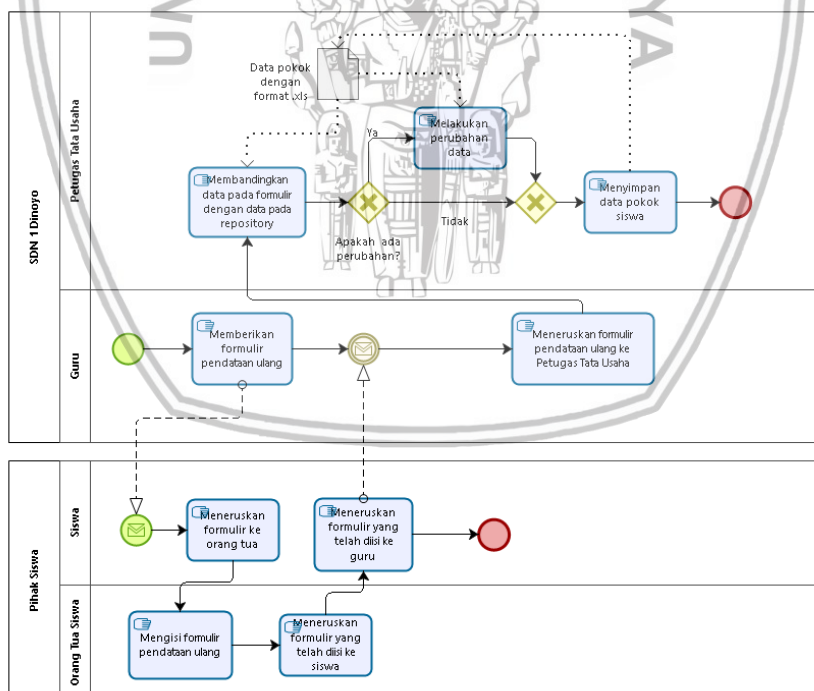
Nama Aktor	Kode Aktivitas	Aktivitas	Deskripsi
Orang Tua Calon Siswa	AA-SIAS-01	Datang ke sekolah	Aktivitas pertama dalam proses bisnis pendaftaran siswa baru. Orang Tua Calon Siswa datang ke sekolah untuk melakukan pendaftaran siswa baru.

Nama Aktor	Kode Aktivitas	Aktivitas	Deskripsi
	AA-SIAS-02	Mengisi buku tamu pendaftaran	Orang Tua Calon Siswa mengisi buku tamu pendaftaran siswa baru yang berupa dokumen cetak.
	AA-SIAS-03	Mengisi formulir pendaftaran siswa baru	Orang Tua Calon Siswa mengisi formulir pendaftaran berupa dokumen cetak yang terdiri dari data pribadi, data ayah kandung, data ibu kandung, data wali, kontak, data priodik, data pribadi, dan data beasiswa.
	AA-SIAS-04	Menyerahkan formulir dan berkas pendukung	Orang Tua Calon Siswa menyerahkan formulir yang telah diisi dan berkas pendukung yang terdiri dari <i>fotocopy</i> akta kelahiran, dan ijazah Taman Kanak-Kanak, kepada Guru.
	AA-SIAS-05	Datang ke sekolah	Ketika jadwal pengumuman, Orang Tua Calon Siswa kembali datang ke sekolah untuk melihat pengumuman siswa yang diterima
	AA-SIAS-06	Melihat pengumuman	Orang Tua Calon Siswa melihat hasil pengumuman siswa yang diterima.
	AA-SIAS-07	Mengumpulkan formulir dan berkas	Guru mengumpulkan semua formulir dan berkas pendaftaran, formulir dan berkas pendaftaran akan menjadi dasar informasi dalam tahap seleksi.
Guru	AA-SIAS-08	Seleksi awal dan merekomendasikan siswa yang diterima	Guru melakukan seleksi awal berdasarkan syarat yang telah ditentukan. Kemudian Guru menentukan siswa yang akan direkomendasikan.

Nama Aktor	Kode Aktivitas	Aktivitas	Deskripsi
	AA-SIAS-09	Menentukan siswa yang diterima	Guru melakukan seleksi terakhir berdasarkan informasi siswa yang direkomendasikan.
Petugas Tata Usaha	AA-SIAS-10	Rekapitulasi data siswa yang diterima	Petugas Tata Usaha melakukan rekapitulasi data siswa yang diterima. Data ini akan menjadi data pokok siswa. Data disimpan pada <i>Microsoft excel</i>
	AA-SIAS-11	Membuat Pengumuman	Petugas Tata Usaha membuat pengumuman nama siswa yang diterima.

#### 4.2.1.2 Proses Bisnis As-Is Pendataan Ulang Data Pokok

Dalam Gambar 4.2 merupakan rangkaian aktivitas pendataan ulang data pokok yang dilakukan oleh Guru, Petugas Tata Usaha SDN 1 Dinoyo, siswa, dan orang tua siswa.



**Gambar 4.2 Proses Bisnis As-Is Pendataan Ulang Data Pokok Siswa**

Uraian proses bisnis pendataan ulang data pokok siswa saat ini yang terdapat pada Gambar 4.2 adalah sebagai berikut:

1. Proses bisnis pendataan ulang data pokok siswa dimulai dengan guru memberikan formulir pendataan ulang kepada siswa.

2. Siswa memberikan formulir pendataan ulang pada orang tua.
3. Orang tua mengisi formulir pendataan ulang.
4. Orang tua mengumpulkan formulir pendataan melalui siswa.
5. Siswa memberikan formulir pendataan kepada guru.
6. Guru menyerahkan formulir pendataan kepada Petugas Tata Usaha.
7. Petugas Tata Usaha akan membandingkan setiap data yang telah disimpan dengan data yang ada pada formulir pendataan, jika terdapat perubahan, maka data akan diperbaharui.

Tabel 4.5 menjelaskan mengenai daftar aktivitas yang terjadi pada proses bisnis pendataan ulang data pokok siswa. Daftar aktivitas terdiri dari nama aktor, kodifikasi aktivitas, nama aktivitas, dan deskripsi singkat aktivitas.

**Tabel 4.5 Daftar Aktivitas Proses Bisnis As-Is Pendataan Ulang Data Pokok Siswa**

<b>Nama Aktor</b>	<b>Kode Aktivitas</b>	<b>Aktivitas</b>	<b>Deskripsi</b>
Guru	AA-SIAS-12	Memberikan formulir pendataan ulang	Guru memberikan formulir pendataan ulang kepada siswa berupa dokumen cetak.
	AA-SIAS-13	Meneruskan formulir pendataan ulang ke Petugas Tata Usaha	Formulir yang telah diisi dikumpulkan melalui guru kelas dan kemudian diberikan kepada petugas tata usaha.
Siswa	AA-SIAS-14	Meneruskan formulir ke Orang Tua	Siswa memberikan formulir kepada orang tua untuk dilakukan pengisian data.
	AA-SIAS-15	Meneruskan formulir yang telah diisi ke Guru	Siswa memberikan formulir yang telah diisi kepada guru kelas.
Orang Tua Siswa	AA-SIAS-16	Mengisi formulir pendataan ulang	Orang Tua Siswa mengisi setiap data pada formulir pendataan ulang.
	AA-SIAS-17	Meneruskan formulir yang telah diisi ke siswa	Formulir yang telah diisi diberikan kepada pihak sekolah melalui siswa.
	AA-SIAS-18	Membandingkan data pada formulir dengan	Petugas Tata Usaha membandingkan setiap data pada formulir dengan ada data



Nama Aktor	Kode Aktivitas	Aktivitas	Deskripsi
Petugas Tata Usaha		data pada repository	yang telah tersimpan pada <i>Microsoft excel</i>
	AA-SIAS-19	Melakukan perubahan data	Petugas Tata Usaha melakukan perubahan data jika ada perbedaan data yang ada pada formulir dengan data pada repository.
	AA-SIAS-20	Menyimpan data pokok siswa	Petugas Tata Usaha menyimpan data pokok siswa.

#### 4.2.1.3 Analisis Masalah

Analisis masalah dilakukan untuk memahami masalah yang terjadi sehingga dapat menjelaskan kebutuhan pengguna, serta untuk mengusulkan solusi terhadap permasalahan. Hasil analisis masalah dilakukan dengan melakukan analisis hasil wawancara dengan Kepala Tata Usaha SDN Dinoyo 1 dan pemodelan proses bisnis *as-is* menggunakan BPMN.

Tabel 4.6 Identifikasi Permasalahan

Kode Identifikasi Pemasalahan	Kode Aktivitas Proses Bisnis <i>as-is</i>	Deskripsi Permasalahan
IM-SIAS-01	AA-SIAS-01	Formulir pendaftaran siswa baru masih berupa dokumen cetak.
IM-SIAS-02	AA-SIAS-03	Data wajib pada formulir tidak semuanya diisi oleh orang tua calon siswa.
IM-SIAS-03	AA-SIAS-07	Berkas pendukung ( <i>fotocopy</i> akte kelahiran dan ijasah Taman Kanak-Kanak) sering tertinggal untuk dikumpulkan.
IM-SIAS-04	AA-SIAS-08	Guru harus membandingkan secara manual data jarak rumah ke sekolah dan umur pada setiap formulir untuk menyeleksi tahap awal berdasarkan data, yang membuat guru menjadi kesulitan.  Tulisan pada formulir yang susah dibaca dan data wajib pada formulir tidak semuanya diisi oleh orang tua calon

Kode Identifikasi Pemasalahan	Kode Aktivitas Proses Bisnis <i>as-is</i>	Deskripsi Permasalahan
		siswa, hal ini menyebabkan sulitnya dalam proses seleksi.
IM-SIAS-05	AA-SIAS-10	Semua data siswa yang diterima, harus dilakukan rekapitulasi untuk menyimpan data pokok siswa. Hal ini membutuhkan waktu yang lama, dan juga petugas tata usaha kesulitan dalam rekapitulasi karena tulisan yang susah dibaca, dan tidak lengkapnya data.
IM-SIAS-06	AA-SIAS-12	Proses pendataan ulang data pokok masih manual dengan mengisikan data pada formulir berupa dokumen cetak.
IM-SIAS-07	AA-SIAS-18	Data pada formulir pendataan tidak lengkap dan sulit dibaca. Hal ini menyebabkan sulitnya dalam melakukan pembaharuan data pokok siswa.
IM-SIAS-08		Petugas Tata Usaha harus membandingkan satu persatu data pokok yang sudah disimpan dengan data yang ada pada formulir pendataan ulang. Jika terjadi perubahan maka Petugas Tata Usaha melakukan pembaharuan. Hal ini dapat menyebabkan Petugas Tata Usaha kebingungan saat melakukan pembaharuan data.
IM-SIAS-09	AA-SIAS-19	Sulitnya mengakses dan mengelola data pokok, karena hanya dapat diakses pada satu tempat saja.

Tabel 4.6 menjelaskan identifikasi permasalahan yang diperoleh dari pemodelan proses bisnis *as-is*. Hasil identifikasi permasalahan terdiri dari kode identifikasi permasalahan, kode aktivitas proses *bisnis as-is*, dan deskripsi singkat permasalahan yang saat ini terjadi.

Informasi mengenai hasil identifikasi permasalahan akan dianalisis menjadi daftar pernyataan masalah (*problem statement*). Pernyataan masalah terdiri dari masalah, aktor yang terpengaruhi, dampak adanya masalah dan juga solusi yang ditawarkan. Tabel 4.7 menjelaskan mengenai pernyataan masalah pada proses bisnis pendaftaran siswa baru.

**Tabel 4.7 Problem Statement 1**

Masalah (The problem of)	<ol style="list-style-type: none"> <li>1. Proses pendaftaran masih manual dengan mengisi formulir dan mengumpulkan berkas yang terkadang hilang dan sulit dibaca.</li> <li>2. Formulir pendaftaran tidak diisi sepenuhnya oleh orang tua siswa.</li> <li>3. Guru harus membandingkan secara manual data jarak rumah ke sekolah dan umur pada setiap formulir untuk menyeleksi tahap awal berdasarkan data, yang membuat guru menjadi kesulitan.</li> <li>4. Berkas pendukung (<i>fotocopy</i> akte kelahiran dan ijasah Taman Kanak-Kanak) sering tertinggal untuk dikumpulkan.</li> <li>5. Terjadi dua kali kerja saat data yang sudah terkumpul harus dilakukan rekapitulasi kembali untuk menyimpan data pokok.</li> </ol>
Mempengaruhi (Affects)	Orang tua calon siswa, Petugas Tata Usaha SDN 1 Dinoyo
Dampak (The impact of which is)	<ol style="list-style-type: none"> <li>1. Data yang diperlukan pihak sekolah untuk pendaftaran siswa baru tidak lengkap.</li> <li>2. Proses seleksi awal mengalami kesulitan karena guru harus membandingkan setiap data jarak rumah sekolah dan umur pada setiap formulir.</li> <li>3. Proses pendaftaran siswa baru menjadi terlambat.</li> <li>4. Berkas pendukung pendaftaran tidak lengkap sehingga orang tua siswa harus dua kali kerja untuk melengkapi berkas pendukung.</li> <li>5. Proses pendataan terjadi perulangan karena formulir registrasi akan di catat kembali pada <i>Microsoft excel</i>.</li> </ol>
Solusi (A successful solution would)	Menyediakan layanan pendaftaran siswa baru yang dapat memastikan data dan berkas pendukung telah diisi sepenuhnya dan menyediakan layanan untuk mengelola data pendaftaran siswa baru.

Tabel 4.8 menjelaskan masalah, aktor yang terpengaruhi, dampak adanya masalah, dan juga solusi yang ditawarkan dalam proses pendataan ulang data pokok siswa.

**Tabel 4.8 Problem Statement 2**

Masalah (The problem of)	<ol style="list-style-type: none"> <li>1. Proses pendataan ulang data pokok masih manual dengan mengisi data pada formulir.</li> </ol>
--------------------------	--

	<ol style="list-style-type: none"> <li>2. Sulitnya mengakses dan mengelola data pokok, karena hanya dapat diakses pada satu tempat saja.</li> <li>3. Data pada formulir pendataan tidak lengkap dan sulit dibaca.</li> <li>4. Petugas Tata Usaha harus membandingkan satu persatu data pokok yang sudah disimpan dengan data yang ada pada formulir pendataan ulang. Jika terjadi perubahan maka Petugas Tata Usaha melakukan pembaharuan.</li> </ol>
Mempengaruhi (Affects)	Orang tua siswa, Petugas Tata Usaha SDN 1 Dinoyo, Guru
Dampak (The impact of which is)	<ol style="list-style-type: none"> <li>1. Proses pendataan ulang data pokok menjadi terlambat.</li> <li>2. Petugas Tata Usaha harus melakukan pengecekan satu persatu data pokok siswa.</li> <li>3. Karena banyaknya data membuat Petugas Tata Usaha kebingungan saat memasukkan data dan bahkan salah memasukkan data.</li> <li>4. Sulitnya mengakses dan mengelola data pokok siswa.</li> </ol>
Solusi (A successful solution would)	Sistem yang dapat menyediakan layanan pendataan ulang data pokok siswa yang dapat digunakan oleh orang tua siswa untuk melakukan pengisian formulir pendataan ulang, dan data disimpan pada suatu <i>repository</i> yang dapat diakses dari mana saja agar Petugas Tata Usaha dapat mengelola data pokok tersebut.

#### 4.2.2 Proses Bisnis *To-Be*

Proses bisnis *to-be* yang dimodelkan pada penelitian ini merupakan proses bisnis usulan yang ditawarkan kepada pemangku kepentingan. Proses bisnis *to-be* diidentifikasi berdasarkan hasil analisis terhadap proses bisnis *as-is*. Kemudian proses bisnis *to-be* akan dimodelkan dengan notasi BPMN untuk menggambarkan perubahan atau tambahan yang diusulkan melalui penelitian ini. Hasil pemodelan proses bisnis *to-be* digunakan sebagai masukan kegiatan analisis kebutuhan.

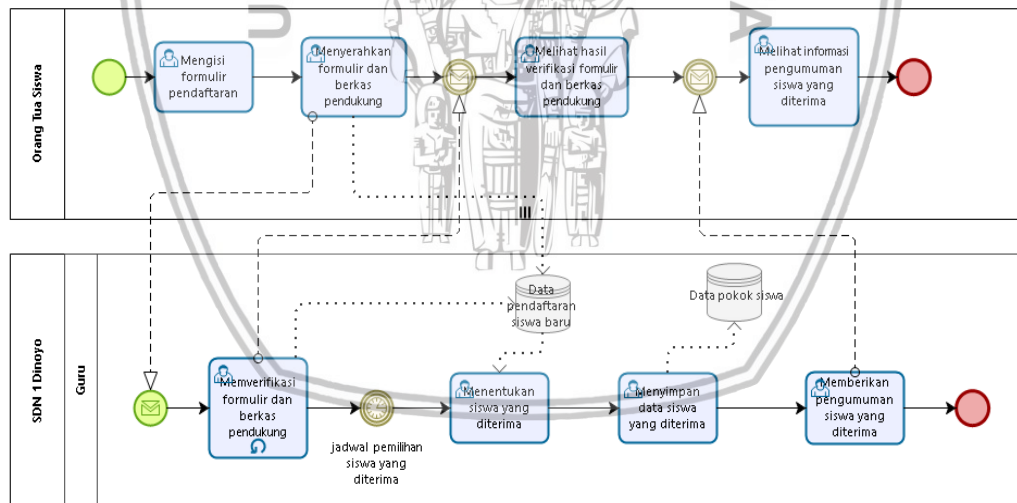
Pada Tabel 4.9 menjelaskan hasil identifikasi proses bisnis usulan (*to-be*). Hasil identifikasi terdiri dari nama proses bisnis, aktor yang terlibat, dan juga deskripsi singkat mengenai proses bisnis yang saat ini berjalan.

Tabel 4.9 Hasil Identifikasi Proses Bisnis *To-Be*

Nama Proses Bisnis	Aktor yang Terlibat	Deskripsi
Pendaftaran Siswa Baru	Orang Tua Calon Siswa, dan Guru.	Rangkaian aktivitas usulan pendaftaran siswa baru yang dilakukan oleh Guru, dan Orang Tua Calon Siswa
Pendataan Ulang Data Pokok Siswa	Orang Tua Siswa, dan Petugas Tata Usaha.	Rangkaian aktivitas usulan pendataan ulang data pokok siswa yang dilakukan oleh Petugas Tata Usaha, dan Orang Tua Siswa

#### 4.2.2.1 Proses Bisnis *To-Be* Pendaftaran Siswa Baru

Dalam Gambar 4.3 merupakan rangkaian proses bisnis usulan terkait aktivitas pendaftaran siswa baru. Pada proses pendaftaran siswa baru, Orang Tua Calon Siswa dapat melakukan pendaftaran secara bersamaan dengan Orang Tua Calon Siswa yang lain tanpa harus saling menunggu proses pendaftaran selesai, sehingga kegiatan pada Orang Tua Calon Siswa dapat dilakukan secara parallel. Untuk melambangkan kegiatan tersebut maka digunakan notasi *multiple instance* bertipe parallel pada *pool* Orang Tua Calon Siswa.



Gambar 4.3 Proses Bisnis *To-Be* Pendaftaran Siswa Baru

Uraian proses bisnis usulan pendaftaran siswa baru yang terdapat pada Gambar 4.3 adalah sebagai berikut:

1. Proses bisnis pendaftaran siswa baru dimulai dengan aktivitas orang tua calon siswa mengisi formulir pendaftaran siswa dan mengumpulkan berkas pendukung, pada tanggal yang telah ditentukan.
2. Guru melakukan verifikasi berkas pendaftaran.
3. Guru menunggu jadwal pemilihan siswa yang diterima, kemudian guru menentukan siswa yang diterima.



4. Sistem menyimpan data siswa yang diterima, data ini akan menjadi data pokok siswa.
5. Sistem kemudian memberlakukan tutup pendaftaran pada jadwal yang telah ditentukan.
6. Sistem menampilkan daftar nama siswa yang diterima.
7. Orang tua calon siswa melihat informasi siswa yang diterima.

Tabel 4.10 menjelaskan mengenai daftar aktivitas pada proses bisnis pendaftaran siswa baru usulan (*to-be*). Daftar aktivitas terdiri dari nama aktor, kodifikasi aktivitas, nama aktivitas, dan deskripsi singkat aktivitas.

**Tabel 4.10 Daftar Aktivitas Proses Bisnis *To-Be* Pendaftaran Siswa Baru**

<b>Nama Aktor</b>	<b>Kode Aktivitas</b>	<b>Aktivitas</b>	<b>Deskripsi</b>
Orang Tua Calon Siswa	AT-SIAS-01	Mengisi formulir pendaftaran	Orang Tua Calon Siswa mengisi formulir pendaftaran siswa baru melalui sistem. Orang Tua Calon Siswa mengisi formulir pendaftaran berupa dokumen cetak yang terdiri dari data pribadi, data ayah kandung, data ibu kandung, data wali, kontak, data priodik, data pribadi, dan data beasiswa melalui sistem.
	AT-SIAS-02	Menyerahkan formulir dan berkas pendukung	Orang Tua Calon Siswa menyerahkan formulir dan berkas pendukung melalui sistem.
	AT-SIAS-03	Melihat hasil verifikasi formulir dan berkas pendukung	Orang Tua Calon Siswa melihat hasil verifikasi formulir dan berkas pendukung melalui sistem.
	AT-SIAS-04	Melihat informasi pengumuman siswa yang diterima	Orang Tua Calon Siswa melihat pengumuman siswa yang diterima melalui sistem.
Guru	AT-SIAS-05	Memverifikasi formulir dan berkas pendukung	Guru melakukan verifikasi formulir dan berkas pendukung melalui sistem.

Nama Aktor	Kode Aktivitas	Aktivitas	Deskripsi
	AT-SIAS-06	Menentukan siswa yang diterima	Guru melakukan pengolahan data dan kemudian menentukan siswa yang diterima melalui sistem.
	AT-SIAS-07	Menyimpan data siswa yang diterima	Guru menyimpan data siswa yang diterima untuk dijadikan data pokok siswa melalui sistem.
	AT-SIAS-08	Memberikan pengumuman siswa yang diterima	Guru menampilkan pengumuman daftar nama siswa yang diterima melalui sistem.

Perubahan aktivitas proses bisnis pendaftaran siswa baru ditunjukkan pada Tabel 4.11.

**Tabel 4.11 Perubahan Aktivitas Proses Bisnis Pendaftaran Siswa Baru**

Unit Bisnis	Proses Bisnis <i>As-Is</i>	Proses Bisnis <i>To-Be</i>	Keterangan
Orang Tua Calon siswa	Datang ke sekolah	-	Dieliminasi
	Mengisi buku tamu pendaftaran siswa baru	-	Dieliminasi
Guru	Mengumpulkan formulir dan berkas	-	Dieliminasi
	Seleksi awal dan merekomendasikan siswa yang diterima	Memverifikasi formulir dan berkas pendukung	Diubah
Petugas Tata Usaha	Rekapitulasi data siswa yang diterima	Menyimpan data siswa yang diterima	Diubah
	Membuat pengumuman	Memberikan pengumuman siswa yang diterima	Diubah

Selain perubahan aktivitas, solusi penggunaan lokasi penyimpanan data terpusat dapat mengeliminasi penggunaan formulir pendaftaran dalam bentuk dokumen cetak yang harus diberikan oleh orang tua calon siswa kepada guru, karena data pendaftaran telah tersimpan dalam penyimpanan data terpusat.

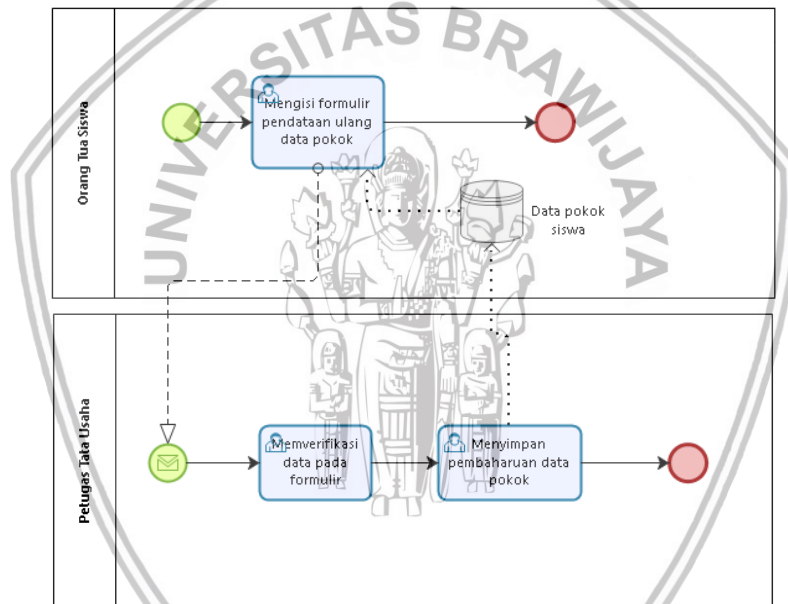
Selain itu perubahan aktivitas terjadi pada petugas tata usaha yang harus merekap setiap data siswa yang telah dinyatakan diterima untuk dijadikan data pokok. Pada proses bisnis usulan, sistem akan menyimpan secara otomatis siswa yang telah dinyatakan diterima oleh guru, sehingga petugas tata usaha tidak perlu merekap satu persatu data siswa tersebut. Kemudian data pokok siswa akan

disimpan pada lokasi penyimpanan data terpusat sehingga dapat memudahkan proses pengelolaan data pokok yang akan dilakukan pada proses bisnis pendataan ulang data pokok siswa. Kemudian orang tua calon siswa tidak perlu datang ke sekolah untuk melihat pengumuman, orang tua calon siswa dapat melihat pengumuman melalui sistem.

#### 4.2.2.2 Proses Bisnis *To-Be* Pendataan Ulang Data Pokok Siswa

Dalam Gambar 4.4 merupakan rangkaian proses bisnis usulan terkait aktivitas pendaftaran siswa baru. Uraian proses bisnis usulan pendataan ulang data pokok siswa yang terdapat pada gambar 4.4 adalah sebagai berikut:

1. Proses bisnis pendataan ulang data pokok siswa dimulai orang tua mengisi formulir pendataan ulang data pokok pada waktu yang ditentukan.
2. Petugas Tata Usaha memverifikasi formulir pendataan ulang.
3. Petugas Tata usaha menyimpan pembaharuan data pokok siswa.



**Gambar 4.4 Proses Bisnis *To-Be* Pendataan Ulang Data Pokok Siswa**

Tabel 4.12 menjelaskan mengenai daftar aktivitas yang terjadi pada proses bisnis pendataan ulang data pokok siswa. Daftar aktivitas terdiri dari nama aktor, kodifikasi aktivitas, nama aktivitas, dan deskripsi singkat aktivitas.

**Tabel 4.12 Daftar Aktivitas Proses Bisnis *To-Be* Pendataan Ulang Data Pokok Siswa**

Nama Aktor	Kode Aktivitas	Aktivitas	Deskripsi
Orang Tua Siswa	AT-SIAS-09	Mengisi formulir	Orang Tua Siswa mengisi setiap data pada formulir pendataan ulang melalui sistem.

		pendataan ulang	
Petugas Tata Usaha	AT-SIAS-10	Memverifikasi data pada formulir	Petugas Tata Usaha melakukan verifikasi data yang telah dimasukkan orang tua siswa melalui sistem.
	AT-SIAS-11	Menyimpan pembaharuan data pokok	Petugas Tata Usaha menyimpan data pokok siswa yang telah diperbaharui melalui sistem.

Perubahan aktivitas proses bisnis pendataan ulang data pokok siswa ditunjukkan pada Tabel 4.13.

**Tabel 4.13 Perubahan Aktivitas Proses Bisnis Pendataan Ulang Data Pokok**

Unit Bisnis	Proses Bisnis <i>As-Is</i>	Proses Bisnis <i>To-Be</i>	Keterangan
Guru	Memberikan formulir pendataan ulang	-	Dieliminasi
	Meneruskan formulir pendataan ulang ke petugas tata usaha	-	Dieliminasi
Siswa	Meneruskan formulir ke orang tua	-	Dieliminasi
	Meneruskan formulir yang telah diisi ke guru	-	Dieliminasi
Orang Tua Siswa	Meneruskan formulir yang telah diisi ke siswa	-	Dieliminasi
Petugas Tata Usaha	Membandingkan data pada formulir dengan data pada repository	Memverifikasi data pada formulir	Diubah
	Melakukan perubahan data	-	Dieliminasi
	Menyimpan data pokok siswa	Menyimpan pembaharuan data pokok	Diubah

Selain perubahan aktivitas, solusi penggunaan lokasi penyimpanan data terpusat dapat mengeliminasi penggunaan dokumen cetak formulir pendataan

ulang yang harus diberikan oleh pihak sekolah kepada orang tua siswa, karena data yang harus diisikan orang tua siswa terdapat pada sistem yang kemudian dapat disimpan pada lokasi penyimpanan data yang terpusat. Petugas tata usaha juga tidak perlu membandingkan setiap data yang diisi pada formulir dengan data yang telah disimpan, karena sistem dapat menyimpan perubahan data yang telah dilakukan oleh orang tua siswa. Selain itu penyimpanan data terpusat dapat membantu pihak sekolah dalam mengakses dan mengelola data pokok siswa tersebut.

### 4.3 Analisis Persyaratan

#### 4.3.1 Identifikasi Tipe Pemangku Kepentingan

Hasil identifikasi tipe pemangku kepentingan berupa daftar contoh kepentingan yang telah dikelompokkan sesuai karakteristik dan hubungan pemangku kepentingan terhadap sistem yang akan dikembangkan dalam penelitian ini. Hasil identifikasi penelitian akan digunakan sebagai informasi untuk kegiatan analisis permasalahan. Hasil identifikasi tipe pemangku kepentingan terdapat pada Tabel 4.14.

**Tabel 4.14 Analisis Tipe Pemangku Kepentingan**

Tipe Pemangku Kepentingan	Deskripsi	Pemangku Kepentingan
Pengguna	Individu yang berinteraksi dengan sistem secara langsung dengan berperan sebagai aktor pada <i>use case</i> .	<ol style="list-style-type: none"> <li>1. Petugas Tata Usaha</li> <li>2. Guru</li> <li>3. Orang Tua Calon Siswa</li> <li>4. Orang Tua Siswa</li> </ol>

#### 4.3.2 Identifikasi Kebutuhan Pemangku Kepentingan dan Pengguna

Identifikasi kebutuhan pemangku kepentingan dan pengguna berisi sekumpulan pernyataan yang berhubungan dengan masalah yang telah diidentifikasi pada tahap analisis masalah. Informasi mengenai kebutuhan pemangku kepentingan dan pengguna diperoleh dengan melakukan analisis terhadap *problem statement* dan hasil wawancara dengan Kepala Sekolah dan Kepala Tata Usaha SDN 1 Dinoyo. Hasil identifikasi kebutuhan pemangku kepentingan dan pengguna digunakan sebagai dasar informasi pada tahap identifikasi fitur.

Hasil Identifikasi kebutuhan pemangku kepentingan dan pengguna dijelaskan pada Tabel 4.15.



**Tabel 4.15 Hasil Identifikasi Kebutuhan Pengguna**

Kode Kebutuhan	Kebutuhan Pengguna	Pemangku Kepentingan	Situasi Saat Ini	Solusi Yang Ditawarkan
N-SIAS-01	Sistem harus menyediakan layanan untuk memasukkan data pendaftaran siswa baru.	Orang tua calon siswa.	Proses pendaftaran siswa baru dilakukan dengan mengambil formulir pendaftaran berupa kertas pada sekolah kemudian orang tua calon siswa mengisi formulir pendaftaran siswa.	Sistem informasi yang menyediakan formulir pendaftaran yang dapat diakses oleh calon orang tua siswa melalui situs web dan memastikan data yang wajib diisi pada formulir pendaftaran telah terisi semua.
N-SIAS-02	Sistem harus menyediakan layanan untuk pengelolaan data pendaftaran siswa baru.	Guru	Data pendaftaran siswa berada pada formulir pendaftaran yang kemudian. Data pendaftaran akan menjadi pertimbangan dalam tahap seleksi. Data pendaftaran siswa yang diterima akan dijadikan data pokok siswa.	Data pendaftaran disimpan dalam satu lokasi penyimpanan yang dapat diakses oleh sistem informasi. Sistem informasi akan memberikan layanan untuk dapat memverifikasi data pendaftaran yang telah diisi orang tua calon siswa. Kemudian sistem dapat menyimpan, dan

Kode Kebutuhan	Kebutuhan Pengguna	Pemangku Kepentingan	Situasi Saat Ini	Solusi Yang Ditawarkan
				menampilkan data pendaftaran siswa baru yang telah diverifikasi.
N-SIAS-03	Sistem harus menyediakan layanan untuk pendataan ulang data pokok siswa.	Orang Tua Siswa, Petugas Tata Usaha.	Proses pendataan data pokok siswa dilakukan dengan mengisi formulir pendataan ulang yang diserahkan pihak sekolah kepada orang tua.	Sistem informasi yang menyediakan formulir pendataan ulang yang dapat diakses orang tua siswa melalui situs web dan memastikan data yang wajib diisi pada formulir pendataan ulang telah terisi semua.
N-SIAS-04	Sistem harus menyediakan layanan untuk mengelola data pokok siswa.	Petugas Tata Usaha, Kepala Sekolah.	Data pokok siswa disimpan pada format dokumen dan hanya dapat dibaca melalui aplikasi <i>Microsoft Excel</i> .	Data pokok siswa akan disimpan dalam satu lokasi penyimpanan yang dapat diakses oleh sistem informasi. Sistem informasi akan memberikan layanan untuk mencatat, mengubah, menyimpan dan menampilkan data pokok siswa.

Kode Kebutuhan	Kebutuhan Pengguna	Pemangku Kepentingan	Situasi Saat Ini	Solusi Yang Ditawarkan
N-SIAS-05	Informasi dan layanan dalam sistem harus dapat diakses sesuai dengan identitas pengguna sistem.	Petugas Tata Usaha, Orang tua calon siswa, orang tua siswa, Guru, Kepala Sekolah.	Tidak ada.	Sistem informasi menyediakan layanan untuk membatasi akses terhadap informasi dan layanan tertentu sesuai dengan hak akses yang dimiliki oleh pengguna sistem.

Hasil identifikasi kebutuhan pemangku kepentingan dan pengguna akan dikelompokkan berdasarkan keterkaitan terhadap analisis permasalahan. Pengelompokan ini berguna untuk menggambarkan bahwa sebuah kebutuhan merupakan solusi untuk menyelesaikan permasalahan tertentu. Pengelompokan fitur dengan kebutuhan pengguna terdapat pada Tabel 4.16.

**Tabel 4.16 Hubungan Kebutuhan Pengguna dengan Analisis Masalah**

Kode Analisis Masalah	Kode Kebutuhan Pengguna
IM-SIAS-01	N-SIAS-01
IM-SIAS-02	
IM-SIAS-03	
IM-SIAS-04	N-SIAS-02
IM-SIAS-05	
IM-SIAS-06	N-SIAS-03
IM-SIAS-07	
IM-SIAS-08	N-SIAS-03
IM-SIAS-09	

### 4.3.3 Identifikasi Pengguna

Identifikasi pengguna merupakan tahap untuk mengetahui individu yang akan menggunakan sistem informasi secara langsung. Informasi mengenai identifikasi pengguna diperoleh berdasarkan contoh pemangku kepentingan yang tergolong di dalam pengguna dan pengembang pada tahap identifikasi tipe pemangku kepentingan. Informasi mengenai identifikasi pengguna digunakan untuk

melakukan identifikasi aktor yang akan dimodelkan ke dalam diagram *use case*. Penjelasan hasil identifikasi pengguna terdapat pada Tabel 4.17.

**Tabel 4.17 Hasil Identifikasi Pengguna**

Tipe Pemangku Kepentingan	Tipe Pengguna	Deskripsi
Pengguna	Petugas Tata Usaha	Pegawai SDN Dinoyo 1 yang berprofesi sebagai petugas tata usaha. Petugas Tata Usaha bertugas untuk mengelola pendaftaran siswa baru, mengelola data pokok siswa dan mengelola administrasi semua kegiatan yang ada di sekolah.
	Guru	Pegawai di SDN Dinoyo 1 yang berprofesi sebagai guru yang bertugas sebagai tenaga pengajar di sekolah. Guru berperan sebagai petugas untuk menyeleksi berkas dan menentukan siswa yang diterima.
	Orang Tua Calon Siswa	Orang tua calon siswa yang mendaftarkan anaknya sebagai siswa SDN 1 Dinoyo.
	Orang Tua Siswa	Orang tua siswa adalah orang tua dari siswa SDN 1 Dinoyo.

#### 4.3.4 Identifikasi Fitur

Identifikasi fitur berguna sebagai representasi beberapa solusi yang ditawarkan untuk memenuhi kebutuhan pengguna. Hasil identifikasi fitur terdiri dari deskripsi singkat dari layanan-layanan yang disediakan oleh sistem untuk memenuhi kebutuhan pengguna dan juga hubungan antara fitur dengan aktivitas proses bisnis usulan (*to-be*). Hasil identifikasi fitur digunakan sebagai informasi untuk identifikasi persyaratan fungsional dan persyaratan nonfungsional. Penjelasan mengenai hasil identifikasi fitur terdapat pada Tabel 4.18.

**Tabel 4.18 Hasil Identifikasi Fitur**

Kode Fitur	Deskripsi	Kode Aktivitas
FEAT-SIAS-01	Sistem dapat mengenali identitas pengguna sistem dan memberikan batasan akses pengguna terhadap informasi dan layanan-layanan yang disediakan oleh sistem berdasarkan hak akses pengguna.	-

Kode Fitur	Deskripsi	Kode Aktivitas
FEAT-SIAS-02	Sistem dapat digunakan untuk menampilkan formulir pendaftaran siswa baru yang kemudian dapat digunakan untuk memasukkan data pendaftaran dan menyediakan layanan untuk memastikan data yang wajib diisi pada formulir pendaftaran telah terisi semua oleh pengguna orang tua calon siswa.	AT-SIAS-01
FEAT-SIAS-03	Sistem dapat digunakan untuk memverifikasi data pendaftaran yang telah diisikan orang tua calon siswa.	AT-SIAS-05
FEAT-SIAS-04	Sistem dapat mengelola data pendaftaran siswa baru yang terdiri dari penyajian semua data, perhitungan jumlah setiap data yang masuk, mengurutkan data siswa berdasarkan tanggal pendaftaran, jarak rumah ke sekolah dan juga umur calon siswa baru.	AT-SIAS-06
FEAT-SIAS-05	Sistem dapat digunakan untuk memonitoring proses seleksi siswa yang terdiri dari melihat informasi status pendaftaran dan melihat informasi siswa yang diterima.	AT-SIAS-03
FEAT-SIAS-06	Sistem dapat digunakan untuk menyimpan data pendaftaran siswa yang diterima untuk dijadikan data pokok siswa.	AT-SIAS-07
FEAT-SIAS-07	Sistem dapat digunakan untuk menampilkan formulir pendataan ulang data pokok siswa yang kemudian dapat digunakan untuk memasukkan data dan menyediakan layanan untuk memastikan data yang wajib diisi pada formulir pendataan ulang telah terisi semua oleh pengguna orang tua siswa.	AT-SIAS-09
FEAT-SIAS-08	Sistem dapat digunakan untuk mengelola data pokok siswa yang terdiri dari penambahan, penyajian, pengubahan dan penghapusan data pokok siswa.	AT-SIAS-10
FEAT-SIAS-09	Sistem dapat digunakan untuk menyimpan data pokok siswa.	AT-SIAS-11

Hasil identifikasi fitur akan dikelompokkan berdasarkan keterkaitan terhadap kebutuhan pengguna yang harus dipenuhi. Pengelompokan ini berguna untuk menggambarkan bahwa sebuah fitur merupakan solusi untuk memenuhi



kebutuhan tertentu. Pengelompokan fitur dengan kebutuhan pengguna terdapat pada Tabel 4.19.

**Tabel 4.19 Hubungan Kebutuhan Pengguna dengan Fitur**

Fitur	Kebutuhan Pengguna
FEAT-SIAS-01	N-SIAS-05
FEAT-SIAS-02	N-SIAS-01
FEAT-SIAS-03	N-SIAS-02
FEAT-SIAS-04	
FEAT-SIAS-05	
FEAT-SIAS-06	
FEAT-SIAS-07	N-SIAS-03
FEAT-SIAS-08	N-SIAS-04
FEAT-SIAS-09	

#### 4.3.5 Persyaratan Fungsional

Persyaratan fungsional merupakan kondisi atau kemampuan yang harus dipenuhi oleh sistem agar fitur sistem dapat berjalan dan dapat memenuhi kebutuhan pengguna. Penjelasan mengenai hubungan antara fitur dengan persyaratan fungsional pada sistem informasi yang akan dikembangkan terdapat pada Tabel 4.20.

**Tabel 4.20 Persyaratan Fungsional**

Kode Fitur	Kode Persyaratan Fungsional	Deskripsi
FEAT-SIAS-01	FR-SIAS-01	Sistem dapat menampilkan halaman otentifikasi pengguna
	FR-SIAS-02	Sistem dapat mengotentifikasi pengguna berdasarkan data pengguna yang tersimpan dalam sistem.
	FR-SIAS-03	Sistem dapat memulai <i>session</i> berdasarkan tipe pengguna yang sedang aktif.
	FR-SIAS-04	Sistem dapat mengganti informasi <i>session</i> berdasarkan tipe pengguna yang sedang aktif.

Kode Fitur	Kode Persyaratan Fungsional	Deskripsi
	FR-SIAS-05	Sistem dapat menghentikan <i>session</i> berdasarkan tipe pengguna yang sedang aktif.
FEAT-SIAS-02	FR-SIAS-06	Sistem dapat menampilkan formulir pendaftaran siswa baru kepada orang tua calon siswa.
	FR-SIAS-07	Sistem dapat digunakan untuk menghasilkan nomor identitas pendaftaran siswa baru.
	FR-SIAS-08	Sistem dapat menampilkan pesan peringatan jika data wajib pada formulir pendaftaran belum diisi.
	FR-SIAS-09	Sistem dapat digunakan untuk menyimpan foto akte kelahiran.
	FR-SIAS-10	Sistem dapat digunakan untuk menyimpan foto ijasah Taman Kanak-Kanak.
	FR-SIAS-11	Sistem dapat digunakan untuk menyimpan semua data pada formulir pendaftaran.
	FR-SIAS-12	Sistem dapat digunakan untuk menampilkan pesan pendaftaran berhasil.
	FR-SIAS-13	Sistem dapat digunakan untuk mencetak nomor pendaftaran.
FEAT-SIAS-03	FR-SIAS-14	Sistem dapat digunakan untuk menampilkan data pendaftaran calon siswa.
	FR-SIAS-15	Sistem dapat digunakan untuk memverifikasi data pendaftaran siswa baru.
FEAT-SIAS-04	FR-SIAS-16	Sistem dapat digunakan untuk memulai pendaftaran.
	FR-SIAS-17	Sistem dapat digunakan untuk menampilkan jumlah pendaftar.
	FR-SIAS-18	Sistem dapat digunakan untuk menampilkan data pendaftaran calon siswa yang telah diverifikasi.

Kode Fitur	Kode Persyaratan Fungsional	Deskripsi
	FR-SIAS-19	Sistem dapat digunakan untuk mengurutkan data pendaftaran calon siswa berdasarkan waktu pendaftaran.
	FR-SIAS-20	Sistem dapat digunakan untuk mengurutkan data pendaftaran calon siswa berdasarkan jarak antara rumah dengan SDN 1 Dinoyo.
	FR-SIAS-21	Sistem dapat digunakan untuk menghitung umur calon siswa.
	FR-SIAS-22	Sistem dapat digunakan untuk mengurutkan data pendaftaran calon siswa berdasarkan umur.
	FR-SIAS-23	Sistem dapat digunakan untuk menyimpan data siswa yang direkomendasikan
	FR-SIAS-24	Sistem dapat digunakan untuk membatalkan data siswa yang direkomendasikan
	FR-SIAS-25	Sistem dapat digunakan untuk menampilkan data siswa yang direkomendasikan.
	FR-SIAS-26	Sistem dapat digunakan untuk menutup pendaftaran.
FEAT-SIAS-05	FR-SIAS-27	Sistem dapat digunakan untuk menampilkan status proses pendaftaran.
	FR-SIAS-28	Sistem dapat digunakan untuk menampilkan informasi nama siswa yang diterima.
FEAT-SIAS-06	FR-SIAS-29	Sistem dapat menyimpan data siswa yang diterima sebagai data pokok.
	FR-SIAS-30	Sistem dapat digunakan untuk menampilkan data siswa yang diterima.
FEAT-SIAS-07	FR-SIAS-31	Sistem dapat menampilkan formulir pendataan ulang data pokok siswa.
	FR-SIAS-32	Sistem dapat digunakan untuk memperbaharui data pokok siswa.

Kode Fitur	Kode Persyaratan Fungsional	Deskripsi
	FR-SIAS-33	Sistem dapat menampilkan pesan peringatan jika data wajib pada formulir pendataan ulang data pokok belum diisi.
FEAT-SIAS-08	FR-SIAS-34	Sistem dapat digunakan untuk menyajikan data jumlah siswa kelas.
	FR-SIAS-35	Sistem dapat digunakan untuk menyajikan data pokok siswa.
	FR-SIAS-36	Sistem dapat digunakan untuk mengubah data pokok siswa.
	FR-SIAS-37	Sistem dapat digunakan untuk menambah data pokok siswa.
	FR-SIAS-38	Sistem dapat digunakan untuk menambah data prestasi siswa.
	FR-SIAS-39	Sistem dapat digunakan untuk menambah data beasiswa siswa.
	FR-SIAS-40	Sistem dapat digunakan untuk memeriksa duplikasi nisl siswa.
	FR-SIAS-41	Sistem dapat digunakan untuk mengubah nisl siswa.
	FR-SIAS-42	Sistem dapat digunakan untuk mencetak data pokok siswa.
	FR-SIAS-43	Sistem dapat digunakan untuk menghapus data pokok siswa
FEAT-SIAS-09	FR-SIAS-44	Sistem dapat menyimpan data pokok siswa yang telah diperbaharui.

#### 4.3.6 Persyaratan Nonfungsional

Identifikasi persyaratan nonfungsional merupakan aspek kualitas atau batasan yang harus dipenuhi oleh sistem. Persyaratan non-fungsional akan menjelaskan persyaratan yang tidak terikat dengan fungsional dari sistem yang berguna untuk mendukung berjalannya sistem yang akan digunakan. Penjelasan mengenai hasil identifikasi persyaratan nonfungsional terdapat pada Tabel 4.21.

**Tabel 4.21 Persyaratan Nonfungsional**

Kode Fitur	Kode persyaratan Nonfungsional	Deskripsi
FEAT-SIAS-01	FR-SIAS-01	

Kode Fitur	Kode persyaratan Nonfungsional	Deskripsi
FEAT-SIAS-02		Kompatibilitas - Sistem harus dapat menjalankan fungsionalitasnya dengan benar minimal menggunakan 3 <i>browser</i> berbeda yang terdapat pada PC maupun pada <i>smartphone</i> .
FEAT-SIAS-03		
FEAT-SIAS-04		
FEAT-SIAS-05		
FEAT-SIAS-06		
FEAT-SIAS-07		
FEAT-SIAS-08		
FEAT-SIAS-09		

#### 4.4 Pemodelan *Use case*

Pemodelan *use case* dilakukan untuk memodelkan kebutuhan dari perilaku sistem yang mengacu kepada hasil dari pemodelan proses bisnis *to-be* dan analisis kebutuhan fungsional. Pada pemodelan *use case* dilakukan identifikasi aktor, identifikasi *use case*, dan spesifikasi untuk masing-masing *use case*.

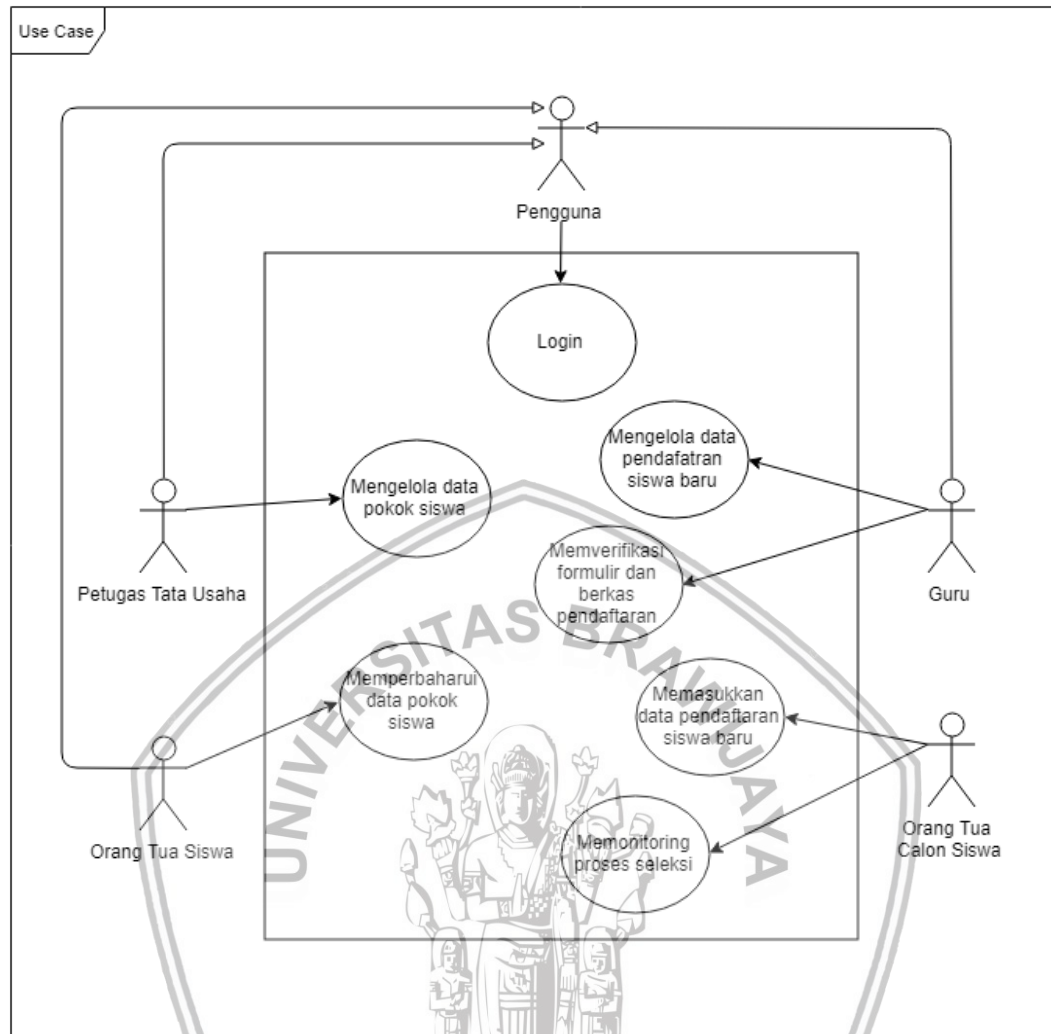
Identifikasi aktor dilakukan dengan mengelompokkan pengguna sistem berdasarkan karakteristik yang dimiliki. Identifikasi *use case* dilakukan dengan menentukan tujuan aktor ketika menggunakan sistem berdasarkan hasil analisis identifikasi kebutuhan pengguna pada tahap analisis persyaratan. Kemudian setiap *use case* akan diberi penjelasan mengenai kondisi yang dibutuhkan sebelum sebuah *use case* dilakukan, kondisi setelah *use case* dilakukan, urutan tahap yang harus dilalui untuk menyelesaikan *use case* dan urutan tahap alternatif jika *use case* memiliki kondisi tertentu. Tahap pemodelan *use case* akan menjadi sumber informasi untuk memodelkan *sequence diagram* pada tahap perancangan.

##### 4.4.1 *Use Case Diagram*

*Use case diagram* memodelkan hubungan aktor dengan *use case*. Informasi mengenai aktor didapatkan berdasarkan hasil identifikasi pengguna pada tahap analisis persyaratan. Sedangkan *use case* diidentifikasi dengan menentukan tujuan aktor dalam menggunakan sistem yang mengacu pada fitur-fitur yang diharapkan pengguna. *Use case diagram* terdapat pada Gambar 4.5.

*Use case* yang teridentifikasi, selanjutnya dihubungkan dengan solusi yang ditawarkan pada aktivitas proses bisnis *to-be* untuk menunjukkan hubungan yang lebih baik dalam penggunaan pemodelan proses bisnis dan *use case* dalam proses pengembangan sistem informasi. Hubungan *use case* dengan aktivitas proses bisnis *to-be* ditunjukkan pada Tabel 4.22.





Gambar 4.5 Use case Diagram

Tabel 4.22 Hubungan Aktivitas Proses Bisnis *To-Be* dengan Use case

Kode Use Case	Use Case	Proses Bisnis	Aktivitas
UC-SIAS-01	Memasukkan data pendaftaran siswa baru	Pendaftaran siswa baru	Mengisi formulir pendaftaran siswa baru
UC-SIAS-02	Mengelola data pendaftaran siswa baru		Menentukan siswa yang diterima
			Menyimpan data siswa yang diterima
UC-SIAS-03	Memonitoring proses seleksi		Memonitoring proses seleksi

Kode Use Case	Use Case	Proses Bisnis	Aktivitas
			Memonitoring proses seleksi
UC-SIAS-04	Memverifikasi formulir dan berkas pendaftaran		Memverifikasi formulir dan berkas pendukung
UC-SIAS-05	Memperbaharui data pokok siswa	Pendataan ulang data pokok siswa	Mengisi formulir pendataan ulang data pokok siswa
UC-SIAS-06	Mengelola data pokok siswa		Menyimpan pembaharuan data pokok siswa
UC-SIAS-07	Login	-	-

Kemudian *use case* yang teridentifikasi dihubungkan dengan fitur-fitur yang teridentifikasi. Hal ini dilakukan untuk mempertegas bahwa hasil pemodelan *use case* dilakukan sesuai dengan hasil analisis persyaratan. Hubungan *use case* dengan fitur ditunjukkan pada Tabel 4.23.

**Tabel 4.23 Hubungan Use case dengan Fitur**

Kode Use Case	Kode Fitur
UC-SIAS-07	FEAT-SIAS-01
UC-SIAS-01	FEAT-SIAS-02
UC-SIAS-02	FEAT-SIAS-04
UC-SIAS-03	FEAT-SIAS-05
UC-SIAS-04	FEAT-SIAS-03
UC-SIAS-05	FEAT-SIAS-07
UC-SIAS-06	FEAT-SIAS-08
	FEAT-SIAS-06
	FEAT-SIAS-09

#### 4.4.2 Deskripsi Aktor

Deskripsi aktor merupakan penjelasan singkat mengenai tanggung jawab aktor yang telah teridentifikasi. Deskripsi aktor menjelaskan secara singkat mengenai tujuan aktor dalam menggunakan sistem informasi yang akan dikembangkan. Deskripsi aktor Pengguna, Petugas Tata Usaha, Guru, Orang Tua Siswa, dan Orang Tua Calon siswa terdapat pada Tabel 4.24.

Tabel 4.24 Deskripsi Aktor

Nama Aktor	Deskripsi
Pengguna	Aktor Pengguna diperankan oleh pegawai Petugas Tata Usaha, Guru, dan Orang Tua Siswa. Aktor pengguna dapat menggunakan sistem dan mengakses sistem informasi sesuai dengan hak akses yang dikenali oleh sistem.
Petugas Tata Usaha	Aktor Petugas Tata Usaha diperankan oleh pegawai Tata Usaha SDN Dinoyo 1 yang akan menggunakan sistem untuk mengelola data pokok siswa.
Guru	Aktor Guru diperankan oleh Guru yang bertindak sebagai penguji untuk menetapkan siswa yang diterima. Aktor Guru dapat menggunakan sistem untuk mengelola data pendaftaran dan verifikasi formulir pendaftaran.
Orang Tua Calon Siswa	Aktor Orang Tua Calon Siswa diperankan oleh orang tua calon siswa yang dapat menggunakan sistem untuk memasukkan data dan berkas pendaftaran dan melihat pengumuman siswa yang diterima.
Orang Tua Siswa	Aktor Orang Tua Siswa diperankan oleh orang tua siswa yang dapat menggunakan sistem untuk memasukkan data pendataan ulang data pokok siswa.

#### 4.4.3 Spesifikasi *Use case*

Spesifikasi *Use case* memuat informasi mengenai penjelasan singkat setiap *use case*, informasi aktor yang menjalankan *use case*, kondisi sebelum dan sesudah *use case* dijalankan, serta tahap-tahap yang dilakukan untuk mencapai *use case*. Tahap-tahap pada spesifikasi *use case* digunakan sebagai panduan pada tahap perancangan.

##### 4.4.3.1 Spesifikasi *Use case Login*

Spesifikasi *use case Login* pada Tabel 4.25 berisi penjelasan mengenai tujuan dan bagaimana aktor Pengguna menggunakan sistem. Di dalam spesifikasi *use case Login* berisi kondisi yang dibutuhkan agar aktor Pengguna dapat menggunakan sistem sesuai dengan tujuannya dan terdapat kondisi apabila *use case* berhasil dilakukan. Selain itu spesifikasi *use case Login* berisi urutan kegiatan yang dilakukan aktor Pengguna ketika menggunakan sistem.

Tabel 4.25 Spesifikasi *Use case Login*

<b>Brief Description</b>	Menjelaskan bagaimana aktor pengguna menggunakan sistem untuk melakukan verifikasi terhadap identitasnya, sehingga dapat mengakses informasi sesuai dengan hak aksesnya.
<b>Actor</b>	Pengguna

<b>Pre-condition</b>	<ul style="list-style-type: none"> <li>• Perangkat yang digunakan untuk mengakses sistem, terhubung dengan internet.</li> <li>• Sistem terhubung dengan <i>server</i>.</li> </ul>
<b>Post-condition</b>	<ul style="list-style-type: none"> <li>• Identitas aktor pengguna teridentifikasi oleh sistem.</li> <li>• Aktor pengguna dinyatakan berhasil masuk ke dalam sistem (<i>logged in</i>).</li> <li>• Sistem menampilkan informasi sesuai hak akses aktor pengguna.</li> </ul>
<b>Basic Flow</b>	<p><b>{Use case dimulai}</b></p> <ol style="list-style-type: none"> <li>1. <i>Use case</i> dimulai ketika aktor Pengguna memasukkan nama dan memasukkan kata sandi.</li> <li>2. Pengguna mengirim nama dan kata sandi ke dalam sistem.</li> <li>3. Sistem mengidentifikasi identitas pengguna. <b>{mengidentifikasi pengguna}</b></li> <li>4. Sistem menyimpan informasi identitas pengguna.</li> <li>5. Sistem menampilkan informasi berdasarkan hak akses aktor pengguna.</li> </ol> <p><b>{Use case selesai}</b></p> <ol style="list-style-type: none"> <li>6. <i>Use case</i> selesai.</li> </ol>
<b>Alternative Flow</b>	<p><b>A1. Menangani kegagalan identifikasi pengguna.</b></p> <p>Pada <b>{mengidentifikasi pengguna}</b>, jika sistem mengalami kegagalan identifikasi pengguna, maka sistem menampilkan pesan bahwa identifikasi pengguna gagal, kemudian <i>use case</i> selesai</p>

#### 4.4.3.2 Spesifikasi *Use case* Memasukkan Data Pendaftaran Siswa Baru

Spesifikasi *use case* Memasukkan Data Pendaftaran Siswa Baru menjelaskan tujuan dan bagaimana aktor Orang Tua Calon Siswa menggunakan sistem. Di dalam *use case* Memasukkan Data Pendaftaran Siswa Baru berisi kondisi yang dibutuhkan agar aktor Orang Tua Calon Siswa menggunakan sistem sesuai dengan tujuannya dan terdapat kondisi apabila *use case* berhasil dilakukan, selain itu berisi urutan kegiatan yang dilakukan aktor Orang Tua Calon Siswa ketika menggunakan sistem. Spesifikasi *Use Case* Memasukkan Data Pendaftaran Siswa Baru terdapat pada Tabel 4.26.

**Tabel 4.26 Spesifikasi *Use Case* Memasukkan Data Pendaftaran Siswa Baru**

<b>Brief Description</b>	Menjelaskan bagaimana aktor Orang Tua Calon Siswa menggunakan sistem untuk memasukkan data pendaftaran siswa baru, berkas-berkas pendukung dan menyimpan pada sistem.
--------------------------	---

<b>Actor</b>	Orang Tua Calon Siswa
<b>Pre-condition</b>	<ul style="list-style-type: none"> <li>Perangkat yang digunakan untuk mengakses sistem, terhubung dengan internet.</li> <li>Sistem terhubung dengan <i>server</i>.</li> </ul>
<b>Post-condition</b>	<ul style="list-style-type: none"> <li>Sistem berhasil menyimpan data dan berkas pendukung pendaftaran siswa baru.</li> <li>Orang Tua Calon Siswa dapat melihat data yang telah disimpan dalam sistem.</li> </ul>
<b>Basic Flow</b>	<p><b>{Use case dimulai}</b></p> <ol style="list-style-type: none"> <li><i>Use case</i> dimulai ketika aktor Orang Tua Calon Siswa memilih untuk memasukkan data pendaftaran siswa baru.</li> <li>Sistem menampilkan formulir pendaftaran siswa baru yang terdiri dari formulir data pribadi, data ayah kandung, data ibu kandung, data wali, kontak, data priodik, prestasi, beasiswa,</li> <li>Orang Tua Calon Siswa mengisikan data pada formulir pendaftaran siswa baru.</li> <li>Orang Tua Calon Siswa memasukkan berkas pendukung.</li> </ol> <p><b>{Memvalidasi kelengkapan data}</b></p> <ol style="list-style-type: none"> <li>Orang Tua Calon Siswa memilih untuk menyimpan data dan berkas pendaftaran siswa baru.</li> </ol> <p><b>{Menyimpan data dan berkas pendaftaran}</b></p> <ol style="list-style-type: none"> <li>Sistem menampilkan pesan penyimpanan data berhasil.</li> </ol> <p><b>{Use case selesai}</b></p> <ol style="list-style-type: none"> <li><i>Use case</i> selesai.</li> </ol>
<b>Alternative Flow</b>	<p><b>A1. Menangani tidak lengkapnya data.</b></p> <p>Pada <b>{memvalidasi kelengkapan data}</b>, jika data wajib pada formulir pendaftaran tidak lengkap diisi oleh aktor Orang Tua Calon Siswa, maka sistem menampilkan pesan bahwa ada data yang belum diisi, dan aktor Orang Tua Calon Siswa tidak dapat memilih untuk menyimpan data dan berkas pendaftaran siswa baru. Kemudian orang tua calon siswa melengkapi data wajib, lalu memilih untuk menyimpan data dan berkas pendaftaran siswa baru. <i>Use case</i> selesai.</p> <p><b>A2. Menangani kegagalan menyimpan data atau berkas pendaftaran.</b></p> <p>Pada <b>{menyimpan data dan berkas pendaftaran}</b>, jika sistem mengalami kegagalan dalam menyimpan data dan berkas pendaftaran, maka sistem akan menampilkan pesan bahwa data atau berkas pendaftaran tidak berhasil disimpan. Kemudian</p>



	sistem menampilkan formulir pendaftaran kembali, dan <i>use case</i> selesai.
--	---

#### 4.4.3.3 Spesifikasi *Use case* Verifikasi Formulir dan Berkas Pendaftaran

Spesifikasi *use case* Verifikasi Formulir dan Berkas Pendaftaran pada Tabel 4.27 berisi penjelasan mengenai tujuan dan bagaimana aktor Guru menggunakan sistem. Di dalam spesifikasi *use case* Verifikasi Formulir dan Berkas Pendaftaran berisi kondisi yang dibutuhkan agar aktor Guru dapat menggunakan sistem sesuai dengan tujuannya dan terdapat kondisi apabila *use case* berhasil dilakukan. Selain itu spesifikasi *use case* Verifikasi Formulir dan Berkas Pendaftaran berisi urutan kegiatan yang dilakukan aktor Guru ketika menggunakan sistem.

**Tabel 4.27 Spesifikasi *Use case* Verifikasi Formulir dan Berkas Pendaftaran**

<b>Brief Description</b>	Menjelaskan bagaimana aktor Guru menggunakan sistem untuk memverifikasi formulir dan berkas pendaftaran yang telah disimpan oleh aktor Orang Tua Calon Siswa.
<b>Actor</b>	Guru
<b>Pre-condition</b>	<ul style="list-style-type: none"> <li>• Perangkat yang digunakan untuk mengakses sistem, terhubung dengan internet.</li> <li>• Sistem terhubung dengan <i>server</i>.</li> <li>• Data pendaftaran siswa telah tersimpan pada sistem.</li> </ul>
<b>Post-condition</b>	<ul style="list-style-type: none"> <li>• Sistem berhasil menyimpan hasil verifikasi data pendaftaran siswa baru.</li> <li>• Aktor Guru dapat melihat hasil verifikasi data pendaftaran siswa baru.</li> </ul>
<b>Basic Flow</b>	<p><b>{Use case dimulai}</b></p> <p><b>{Verifikasi Formulir dan Berkas Pendaftaran}</b></p> <ol style="list-style-type: none"> <li>1. <i>Use case</i> dimulai ketika aktor Guru memilih untuk melihat data pendaftaran siswa baru yang masuk.</li> <li>2. Sistem menampilkan informasi data calon siswa yang telah terdaftar secara urut berdasarkan tanggal calon siswa mendaftar.</li> <li>3. Guru memilih untuk memverifikasi data pendaftaran yang masuk.</li> </ol> <p><b>{Menyimpan verifikasi data dan berkas pendaftaran}</b></p> <ol style="list-style-type: none"> <li>4. Sistem merubah status pendaftaran menjadi terverifikasi.</li> <li>5. Sistem menampilkan pesan verifikasi berhasil.</li> <li>6. <i>Use case</i> selesai.</li> </ol>

<b>Alternative Flow</b>	<p><b>A1. Menangani Kegagalan Penyimpanan Verifikasi Data dan Berkas Pendaftaran.</b></p> <p>Pada {menyimpan verifikasi data dan berkas pendaftaran}, jika sistem mengalami kegagalan dalam menyimpan data, maka sistem akan menampilkan pesan bahwa data tidak berhasil disimpan. <i>use case</i> selesai.</p>
-------------------------	---

#### 4.4.3.4 Spesifikasi *Use case* Mengelola Data Pendaftaran Siswa Baru

Spesifikasi *use case* Mengelola Data Pendaftaran Siswa Baru pada Tabel 4.28 berisi penjelasan mengenai tujuan dan bagaimana aktor Guru menggunakan sistem. Di dalam spesifikasi *use case* Mengelola Data Pendaftaran Siswa Baru berisi kondisi yang dibutuhkan agar aktor Guru dapat menggunakan sistem dengan tujuannya, kondisi apabila *use case* berhasil dilakukan, dan urutan kegiatan yang dilakukan aktor Guru ketika menggunakan sistem.

**Tabel 4.28 Spesifikasi *Use case* Mengelola Data Pendaftaran Siswa Baru**

<b>Brief Description</b>	Menjelaskan bagaimana aktor Guru menggunakan sistem untuk melihat data pendaftaran siswa baru yang telah terverifikasi, mengurutkan data berdasarkan tanggal pendaftaran, jarak rumah calon siswa dengan sekolah dan umur calon siswa. Guru juga dapat menggunakan sistem untuk menyimpan data calon siswa ke dalam daftar siswa yang diterima.
<b>Actor</b>	Guru
<b>Pre-condition</b>	<ul style="list-style-type: none"> <li>• Perangkat yang digunakan untuk mengakses sistem, terhubung dengan internet.</li> <li>• Sistem terhubung dengan <i>server</i>.</li> <li>• Data pendaftaran siswa telah tersimpan pada sistem.</li> </ul>
<b>Post-condition</b>	<ul style="list-style-type: none"> <li>• Sistem berhasil menyimpan hasil pengelolaan data pendaftaran siswa baru.</li> <li>• Aktor Guru dapat melihat hasil pengelolaan data pendaftaran siswa baru.</li> </ul>
<b>Basic Flow</b>	<p><b>{Use case dimulai}</b></p> <p><b>{Melihat data pendaftaran yang telah diverifikasi}</b></p> <ol style="list-style-type: none"> <li>1. <i>Use case</i> dimulai ketika aktor Guru memilih untuk melihat data pendaftaran siswa baru yang telah terverifikasi.</li> <li>2. Sistem menampilkan informasi data calon siswa yang telah terdaftar secara urut berdasarkan tanggal calon siswa mendaftar.</li> <li>3. Lakukan <i>subflow</i> <b>Mengurutkan Data Berdasarkan Jarak Rumah.</b></li> </ol>

	<ol style="list-style-type: none"> <li>Lakukan <i>subflow</i> <b>Mengurutkan Data Berdasarkan Umur Siswa.</b></li> <li>Lakukan <i>subflow</i> <b>Merubah Data Pendaftaran.</b></li> <li>Lakukan <i>subflow</i> <b>Menyimpan Data Siswa yang Diterima.</b></li> <li>Lakukan <i>subflow</i> <b>Menampilkan Pengumuman Siswa yang Diterima.</b></li> <li><i>Use case</i> selesai.</li> </ol>
<b>Alternative Flow</b>	<p><b>A1. Menangani Kegagalan Penyimpanan Data Siswa yang Diterima.</b></p> <p>Pada {menyimpan data siswa yang diterima}, jika sistem mengalami kegagalan dalam menyimpan data, maka sistem akan menampilkan pesan bahwa data tidak berhasil disimpan. <i>use case</i> selesai.</p>
<b>Subflow</b>	<p><b>S1. Mengurutkan Data Berdasarkan Jarak Rumah.</b></p> <ol style="list-style-type: none"> <li>Aktor Guru memilih untuk mengurutkan data berdasarkan jarak rumah.</li> <li>Sistem melakukan proses pemuatan data pendaftaran.</li> <li>Sistem menampilkan menampilkan data pendaftaran berdasarkan jarak terdekat rumah calon siswa ke sekolah.</li> </ol> <p><b>S2. Mengurutkan Data Berdasarkan Umur Calon Siswa.</b></p> <ol style="list-style-type: none"> <li>Aktor Guru memilih untuk mengurutkan data berdasarkan umur calon siswa.</li> <li>Sistem melakukan proses pemuatan data pendaftaran.</li> <li>Sistem menampilkan menampilkan data pendaftaran berdasarkan umur terkecil calon siswa.</li> </ol> <p><b>S3. Merubah Data Siswa yang Diterima.</b></p> <ol style="list-style-type: none"> <li>Aktor Guru memilih untuk merubah data pendaftar yang dipilih untuk dirubah.</li> <li>Sistem menampilkan data pendaftaran.</li> <li>Guru melakukan perubahan data.</li> <li>Guru memilih untuk menyimpan data pendaftaran yang telah dirubah.</li> <li>Sistem menyimpan data pendaftaran. <b>{Menyimpan data siswa yang diterima}</b></li> </ol> <p><b>S4. Menyimpan Data Siswa yang Diterima.</b></p> <ol style="list-style-type: none"> <li>Aktor Guru memilih untuk menyimpan data siswa yang dipilih untuk menjadi data siswa yang diterima.</li> <li>Sistem menyimpan data siswa yang diterima sebagai data pokok siswa.</li> <li>Sistem menampilkan pesan penyimpanan data berhasil.</li> </ol>

	<p><b>{Menyimpan data siswa yang diterima}</b></p> <p><b>S5. Menampilkan Pengumuman Siswa yang Diterima</b></p> <ol style="list-style-type: none"> <li>1. Aktor Guru memilih untuk menampilkan pengumuman siswa yang diterima.</li> <li>2. Sistem menampilkan pengumuman siswa yang diterima.</li> </ol>
--	--

#### 4.4.3.5 Spesifikasi *Use Case* Memonitoring Proses Seleksi

Spesifikasi *use case* Memonitoring Proses Seleksi menjelaskan tujuan dan bagaimana aktor Orang Tua Calon Siswa menggunakan sistem. Di dalam *use case* Memonitoring Proses Seleksi berisi kondisi yang dibutuhkan agar aktor Orang Tua Calon Siswa menggunakan sistem sesuai dengan tujuannya, kondisi apabila *use case* berhasil dilakukan, dan berisi urutan kegiatan yang dilakukan aktor Orang Tua Calon Siswa ketika menggunakan sistem. Spesifikasi *use case* Membaca Pengumuman Siswa yang Diterima terdapat pada Tabel 4.29.

**Tabel 4.29 Spesifikasi *Use Case* Memonitoring Proses Seleksi**

<b>Brief Description</b>	Menjelaskan bagaimana aktor Orang Tua Calon Siswa menggunakan sistem untuk memonitoring proses seleksi
<b>Actor</b>	Orang Tua Calon Siswa
<b>Pre-condition</b>	<ul style="list-style-type: none"> <li>• Perangkat yang digunakan untuk mengakses sistem, terhubung dengan internet.</li> <li>• Sistem terhubung dengan <i>server</i>.</li> <li>• Data siswa yang diterima telah disimpan pada sistem.</li> </ul>
<b>Post-condition</b>	<ul style="list-style-type: none"> <li>• Aktor Orang Tua Calon Siswa mendapatkan informasi mengenai siswa yang diterima yang disimpan pada sistem.</li> </ul>
<b>Basic Flow</b>	<p><b>{Use case dimulai}</b></p> <ol style="list-style-type: none"> <li>1. <i>Use case</i> dimulai ketika aktor Orang Tua Calon Siswa memilih untuk memonitoring proses seleksi.</li> <li>2. Lakukan <i>subflow</i> <b>Melihat Status Pendaftaran</b>.</li> <li>3. Lakukan <i>subflow</i> <b>Melihat Informasi Siswa yang Diterima</b>.</li> <li>4. <i>Use Case</i> selesai.</li> </ol> <p><b>{Use case selesai}</b></p>
<b>Subflow</b>	<p><b>S1. Melihat Status Pendaftaran.</b></p> <ol style="list-style-type: none"> <li>1. Aktor Orang Tua Calon Siswa memilih untuk melihat status pendaftaran.</li> <li>2. Sistem menampilkan halaman untuk memasukkan nomor pendaftaran dan tanggal lahir.</li> <li>3. Orang Tua Siswa memasukkan nomor pendaftaran dan tanggal lahir.</li> </ol>

	<p>4. Sistem menampilkan status pendaftaran.</p> <p><b>S2. Melihat Informasi Siswa yang Diterima.</b></p> <ol style="list-style-type: none"> <li>1. Aktor Orang Tua Calon Siswa memilih untuk melihat informasi siswa yang diterima.</li> <li>2. Sistem menampilkan halaman untuk memasukkan nomor pendaftaran dan tanggal lahir.</li> <li>3. Orang Tua Siswa memasukkan nomor pendaftaran dan tanggal lahir.</li> <li>4. Sistem menampilkan pesan calon siswa diterima.</li> </ol>
--	---

#### 4.4.3.6 Spesifikasi Use Case Memasukkan Data Pokok Siswa

Spesifikasi *use case* Memasukkan Data Pokok Siswa pada Tabel 4.30 berisi penjelasan mengenai tujuan dan bagaimana aktor Orang Tua Siswa menggunakan sistem. Di dalam spesifikasi *use case* Memasukkan Data Pokok Siswa berisi kondisi yang dibutuhkan agar aktor Orang Tua Siswa dapat menggunakan sistem dengan tujuannya, kondisi apabila *use case* berhasil dilakukan, dan urutan kegiatan yang dilakukan aktor Orang Tua Siswa ketika menggunakan sistem.

**Tabel 4.30 Spesifikasi Use Case Memasukkan Data Pokok Siswa**

<b>Brief Description</b>	Menjelaskan bagaimana aktor Orang Tua Siswa menggunakan sistem untuk memasukkan data pokok dan menyimpan pada sistem.
<b>Actor</b>	Orang Tua Siswa
<b>Pre-condition</b>	<ul style="list-style-type: none"> <li>• Perangkat yang digunakan untuk mengakses sistem, terhubung dengan internet.</li> <li>• Sistem terhubung dengan <i>server</i>.</li> <li>• Aktor Orang Tua Siswa berhasil masuk ke dalam sistem (<i>logged in</i>).</li> </ul>
<b>Post-condition</b>	<ul style="list-style-type: none"> <li>• Sistem berhasil menyimpan data pokok pada sistem.</li> <li>• Orang Tua Calon Siswa dapat melihat hasil data pokok yang telah disimpan dalam sistem.</li> </ul>
<b>Basic Flow</b>	<p><b>{Use case dimulai}</b></p> <ol style="list-style-type: none"> <li>1. <i>Use case</i> dimulai ketika aktor Orang Tua Calon Siswa memilih untuk memasukkan data pokok.</li> <li>2. Sistem menampilkan formulir pendataan ulang data pokok yang terdiri dari formulir data pribadi, data ayah kandung, data ibu kandung, data wali, kontak, data priodik, prestasi, beasiswa,</li> <li>3. Orang Tua Siswa mengisi data pada formulir pendataan ulang.</li> </ol> <p><b>{Memvalidasi kelengkapan data}</b></p>



	<p>4. Orang Tua Calon Siswa memilih untuk menyimpan data pokok.  <b>{Menyimpan data pokok}</b></p> <p>5. Sistem menampilkan pesan penyimpanan data berhasil.  <b>{Use case selesai}</b></p> <p>6. <i>Use case</i> selesai.</p>
<b>Alternative Flow</b>	<p><b>A1. Menangani tidak lengkapnya data.</b></p> <p>Pada <b>{memvalidasi kelengkapan data}</b>, jika data wajib pada formulir pendataan ulang tidak lengkap diisi oleh aktor Orang Tua Siswa, maka sistem menampilkan pesan bahwa ada data yang belum diisi, dan aktor Orang Tua Siswa tidak dapat memilih untuk menyimpan data pokok. Kemudian orang tua calon siswa melengkapi data wajib, lalu memilih untuk menyimpan data. <i>Use case</i> selesai.</p> <p><b>A2. Menangani kegagalan menyimpan data pokok.</b></p> <p>Pada <b>{menyimpan data pokok}</b>, jika sistem mengalami kegagalan dalam menyimpan data pokok, maka sistem akan menampilkan pesan bahwa data pokok tidak berhasil disimpan. Kemudian sistem menampilkan formulir pendataan ulang kembali, dan <i>use case</i> selesai.</p>

#### 4.4.3.7 Spesifikasi *Use case* Mengelola Data Pokok Siswa

Spesifikasi *use case* Mengelola Data Pokok Siswa menjelaskan tujuan dan bagaimana aktor Petugas Tata Usaha menggunakan sistem. Di dalam *use case* Mengola Data Pokok Siswa berisi kondisi yang dibutuhkan agar aktor Petugas Tata Usaha menggunakan sistem sesuai dengan tujuannya, kondisi apabila *use case* berhasil dilakukan, dan berisi urutan kegiatan yang dilakukan aktor Petugas Tata Usaha ketika menggunakan sistem. Spesifikasi *Use Case* Mengelola Data Pokok Siswa Tabel 4.31.

**Tabel 4.31 Spesifikasi *Use case* Mengelola Data Pokok Siswa**

<b>Brief Description</b>	Menjelaskan bagaimana aktor Petugas Tata Usaha menggunakan sistem untuk melihat data pokok siswa, mengubah data pokok siswa, menghapus data pokok siswa, dan menambahkan data pokok siswa.
<b>Actor</b>	Petugas Tata Usaha
<b>Pre-condition</b>	<ul style="list-style-type: none"> <li>• Perangkat yang digunakan untuk mengakses sistem, terhubung dengan internet.</li> <li>• Sistem terhubung dengan <i>server</i>.</li> <li>• Data pokok siswa telah tersimpan pada sistem.</li> </ul>

<b>Post-condition</b>	<ul style="list-style-type: none"> <li>Sistem berhasil menyimpan hasil pengelolaan data pokok siswa.</li> <li>Aktor Petugas Tata Usaha dapat melihat hasil pengelolaan data pendaftaran siswa baru.</li> </ul>
<b>Basic Flow</b>	<p><b>{Use case dimulai}</b></p> <p><b>{Menampilkan detail data pokok siswa }</b></p> <ol style="list-style-type: none"> <li>1. <i>Use case</i> dimulai ketika aktor Petugas Tata Usaha memilih untuk melihat data pokok siswa yang telah tersimpan.</li> <li>2. Sistem menampilkan daftar kelas.</li> <li>3. Petugas Tata Usaha memilih daftar kelas.</li> <li>4. Sistem menampilkan data pokok siswa.</li> <li>5. Lakukan <i>subflow</i> <b>Verifikasi Data Pokok Siswa</b>.</li> <li>6. Lakukan <i>subflow</i> <b>Merubah Data Pokok Siswa</b>.</li> <li>7. Lakukan <i>subflow</i> <b>Menghapus Data Pokok Siswa</b>.</li> <li>8. Lakukan <i>subflow</i> <b>Menambah Data Pokok Siswa</b>.</li> <li>9. Lakukan <i>subflow</i> <b>Mencetak Data Pokok Siswa</b>.</li> <li>10. <i>Use case</i> selesai.</li> </ol>
<b>Alternative Flow</b>	<p><b>A1. Menangani Kegagalan Penyimpanan Data Pokok Siswa.</b></p> <p>Pada <b>{menyimpan data pokok siswa}</b>, jika sistem mengalami kegagalan dalam menyimpan data, maka sistem akan menampilkan pesan bahwa data tidak berhasil disimpan. <i>use case</i> selesai.</p> <p><b>A2. Menangani Kegagalan Penghapusan Data Pokok Siswa.</b></p> <p>Pada <b>{Menghapus data pokok siswa}</b>, jika sistem mengalami kegagalan dalam menghapus data, maka sistem akan menampilkan pesan bahwa data tidak berhasil dihapus. <i>use case</i> selesai.</p>
<b>Subflow</b>	<p><b>S1. Verifikasi Data Pokok Siswa</b></p> <ol style="list-style-type: none"> <li>1. Petugas Tata Usaha memilih untuk verifikasi Data Pokok Siswa.</li> <li>2. Petugas Tata Usaha memilih data pokok siswa yang ingin diverifikasi.</li> <li>3. Sistem menyimpan data pokok siswa yang telah berubah.</li> </ol> <p><b>{Menyimpan data pokok siswa}</b></p> <p><b>S2. Merubah Data Pokok Siswa</b></p> <ol style="list-style-type: none"> <li>1. Petugas Tata Usaha memilih untuk merubah Data Pokok Siswa.</li> <li>2. Petugas Tata Usaha memilih data pokok siswa yang ingin dirubah.</li> </ol>

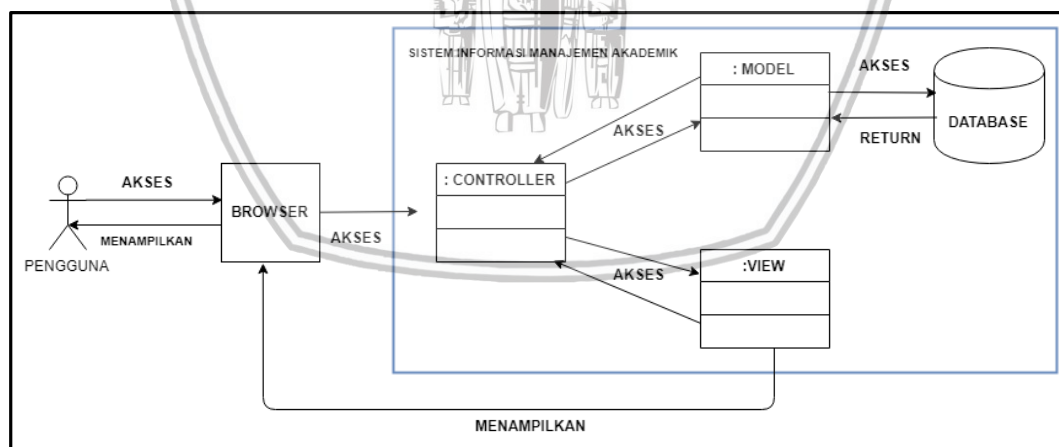
	<ol style="list-style-type: none"> <li>3. Sistem menampilkan data pokok siswa.</li> <li>4. Petugas Tata Usaha merubah data pokok siswa.</li> <li>5. Petugas Tata Usaha menyimpan data pokok siswa yang telah berubah.</li> </ol> <p><b>{Menyimpan data pokok siswa}</b></p> <p><b>S3. Menghapus Data Pokok Siswa.</b></p> <ol style="list-style-type: none"> <li>1. Petugas Tata Usaha memilih untuk menghapus data pokok siswa.</li> <li>2. Petugas Tata Usaha memilih data pokok siswa yang ingin dihapus.</li> <li>3. Sistem melakukan proses menghapus data pokok siswa.</li> </ol> <p><b>{Menghapus data pokok siswa}</b></p> <p><b>S4. Menambah Data Pokok Siswa.</b></p> <ol style="list-style-type: none"> <li>1. Petugas Tata Usaha memilih untuk menambah data pokok siswa.</li> <li>2. Petugas Tata Usaha mengisi formulir penambahan data pokok siswa.</li> <li>3. Sistem melakukan proses penyimpanan data pokok siswa.</li> </ol> <p><b>{Menyimpan data pokok siswa}</b></p> <p><b>S5. Mencetak Data Pokok Siswa.</b></p> <ol style="list-style-type: none"> <li>1. Petugas Tata Usaha memilih untuk mencetak data pokok siswa.</li> <li>2. Sistem melakukan proses pencetakan data pokok siswa.</li> <li>3. Sistem menampilkan hasil cetakan data pokok siswa pada file bertipe <i>Microsoft Word</i>.</li> </ol>
--	--

## BAB 5 PERANCANGAN

Pada bab ini menjelaskan mengenai tahapan perancangan sistem yang merupakan penerapan dari metode *waterfall* pada tahap *system and software design*. Perancangan dibuat berdasarkan hasil dari tahap analisis kebutuhan. Bab perancangan berisi pembahasan mengenai hasil pemodelan dengan menggunakan UML sebagai notasi pemodelan. Hasil pemodelan yang dijelaskan pada bab ini di antaranya adalah perancangan arsitektur sistem, *sequence diagram*, diagram kelas analisis, diagram kelas perancangan, perancangan basis data yang didokumentasikan ke dalam PDM, perancangan algoritme yang didokumentasikan ke dalam *pseudocode*, dan perancangan antarmuka pengguna.

### 5.1 Perancangan Arsitektur Sistem

Perancangan arsitektur sistem digunakan untuk menjelaskan komponen struktur utama dari sistem dan hubungan antar komponen-komponen tersebut. Perancangan arsitektur sistem terdapat pada Gambar 5.1. Pada Gambar 5.1 menjelaskan mengenai arsitektur sistem yang dibangun menggunakan kerangka kerja *CodeIgniter* dengan menerapkan pola *Model-View-Controller* (MVC). Kelas *Controller* merupakan semua kelas turunan dari kelas abstract *CI\_Controller*, sedangkan Kelas *Model* merupakan semua kelas turunan dari kelas abstract *CI\_Model*. Kelas view adalah halaman antarmuka pengguna. Pengguna dapat mengakses sistem melalui *browser*, kemudian *browser* memproses kelas model dan kelas view sistem melalui kelas *controller*. Sedangkan *database* sistem dapat diakses melalui kelas model.



Gambar 5.1 Arsitektur Sistem Informasi Manajemen Akademik

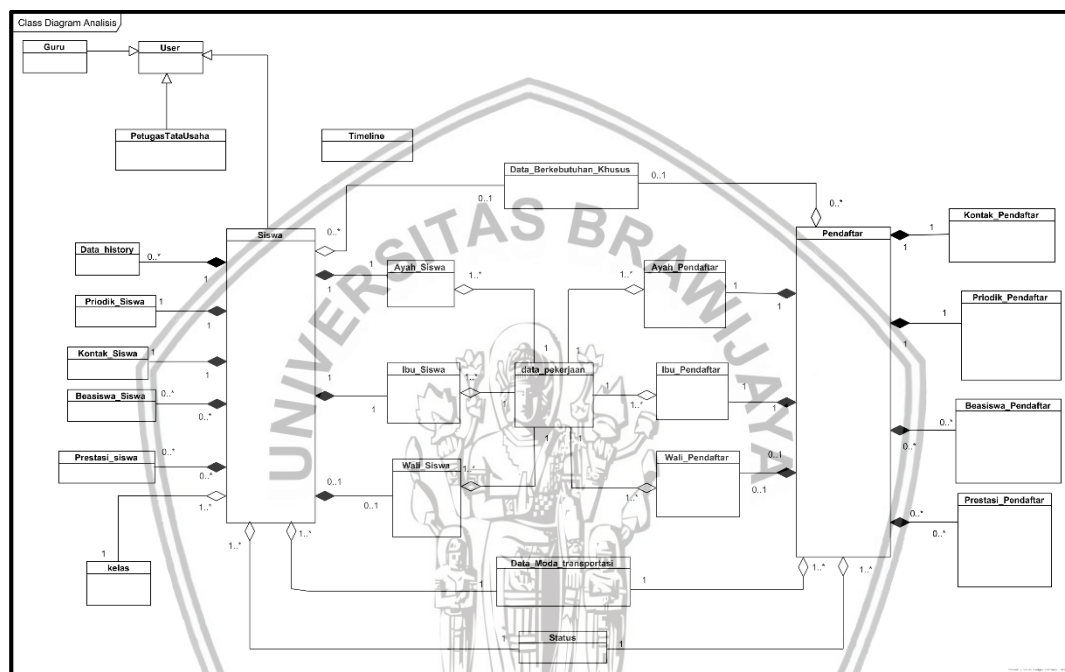
### 5.2 Pemodelan Objek

Pemodelan objek dilakukan untuk mendokumentasikan kumpulan *class* atau *object* dari sistem yang akan dikembangkan dan hubungan dari setiap *class* atau *object* tersebut. Pemodelan objek divisualisasikan melalui diagram kelas. Pada bagian ini menjelaskan mengenai diagram kelas analisis dan diagram kelas perancangan. Diagram kelas analisis berguna sebagai panduan untuk membuat

perancangan basis data. Sedangkan diagram kelas perancangan merupakan visualisasi kelas dan hubungan di antara setiap kelas sebagai panduan untuk melakukan pengembangan sistem.

### 5.2.1 Diagram Kelas Analisis

Diagram kelas analisis yang ditunjukkan dalam Gambar 5.2 berisi visualisasi hubungan antar kelas sebagai representasi *object* yang menjadi komponen penyusun sistem yang akan dikembangkan. Setiap kelas pada diagram didapatkan berdasarkan setiap *use case diagram* yang diidentifikasi berdasarkan hasil analisis yang telah dilakukan.



Gambar 5.2 Diagram Kelas Analisis

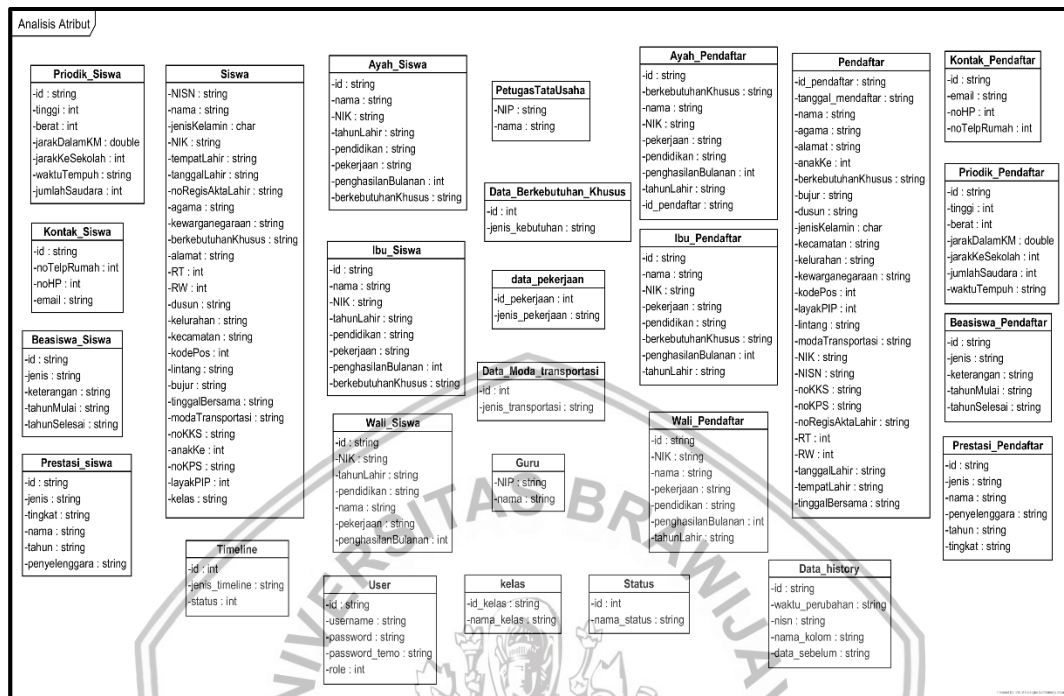
Diagram kelas analisis menunjukkan bahwa terdapat 26 kelas dengan masing-masing kelas memiliki hubungan dengan kelas yang lain. Diagram kelas analisis akan digunakan sebagai panduan pemodelan kelas perancangan dan perancangan basis data.

Pada kegiatan pendaftaran siswa baru terdapat beberapa kelas yang merepresentasikan objek dalam kegiatan tersebut diantaranya yaitu kelas Pendaftar merupakan representasi objek orang tua siswa yang mendaftarkan anaknya. Kelas Pendaftar memiliki hubungan *composite* terhadap kelas Ayah\_Pendaftar yang merepresentasikan objek ayah yang mendaftarkan anaknya. Selain itu kelas Pendaftar memiliki hubungan *composite* terhadap kelas Ibu\_Pendaftar, Wali\_Pendaftar, Kontak\_Pendaftar, Prdiok\_Pendaftar, Beasiswa\_Pendaftar, Prestasi\_Pendaftar.

Sedangkan pada kegiatan pendataan ulang data pokok siswa, terdapat beberapa kelas yang merepresentasikan kegiatan tersebut yaitu kelas Siswa merupakan representasi objek siswa memiliki hubungan *composite* terhadap kelas



Ayah\_Siswa yang merepresentasikan objek ayah dari setiap siswa. Selain itu kelas Siswa juga memiliki hubungan *composite* terhadap kelas Ibu\_Siswa, Wali\_Siswa, Priodik\_Siswa, Kontak\_Siswa, Beasiswa\_Siswa, Prestasi\_Siswa.



Gambar 5.3 Analisis Atribut

Kelas data\_pekerjaan yang merepresentasikan objek pekerjaan memiliki hubungan agregasi terhadap kelas Ayah\_Siswa, Ibu\_Siswa, Wali\_Siswa, Ayah\_Pendaftar, M\_Pendaftar dan Wali\_Pendaftar. Kelas Data\_Moda\_Transportasi merepresentasikan objek moda transportasi yang digunakan siswa atau pendaftar ke sekolah. Kelas Data\_Moda\_Transportasi memiliki hubungan agregasi terhadap kelas Siswa dan jug kelas Pendaftar.

Pada Gambar 5.3 menggambarkan atribut yang dimiliki oleh setiap kelas analisis. Pada kegiatan pendaftaran siswa baru, kelas Pendaftar yang merupakan representasi objek orang tua siswa yang mendaftarkan anaknya memuat atribut id\_pendaftar yang bertipe data string sebagai identitas objek. Kelas Ayah\_Pendaftar, Ibu\_Pendaftar, dan Wali\_Pendaftar memiliki atribut id yang menjadi identitas dari setiap objek. Sedangkan kelas Kontak\_Pendaftar, Priodik\_Pendaftar, Beasiswa\_Pendaftar, dan Prestasi\_Pendaftar memiliki atribut id yang merupakan atribut referensi dari kelas Pendaftar sebagai identitas objek.

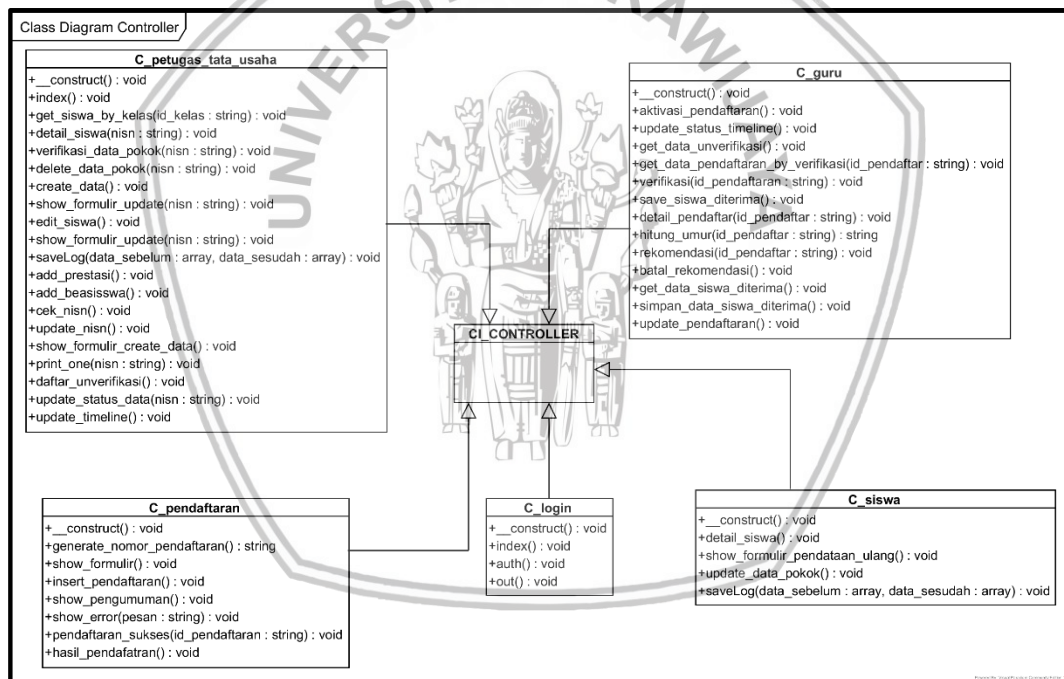
Sedangkan pada kegiatan pendataan ulang data pokok siswa, kelas Siswa yang merupakan representasi objek siswa memuat atribut identitas siswa dengan nomor nisan yang bertipe data string sebagai identitas objek. Kelas Ayah\_Siswa, Ibu\_Siswa, dan Wali\_Siswa memiliki atribut id yang menjadi identitas dari setiap objek. Sedangkan kelas Kontak\_Siswa, Priodik\_Siswa, Beasiswa\_Siswa, dan Prestasi\_Siswa memiliki atribut id yang merupakan atribut referensi dari kelas Siswa sebagai identitas objek.

## 5.2.2 Diagram Kelas Perancangan

Diagram kelas perancangan merupakan hasil pengembangan diagram kelas analisis yang telah dibuat, yang ditunjukkan pada Gambar 5.2. Diagram kelas perancangan dibuat dengan menyesuaikan objek-objek yang teridentifikasi pada diagram kelas analisis. Diagram kelas perancangan terdiri dari subkelas *controller* dan subkelas *model*. Masing-masing subkelas hanya memiliki fungsi (*behavior*). Fungsi yang teridentifikasi pada masing-masing subkelas berdasarkan persyaratan fungsional yang telah teridentifikasi pada Tabel 4.20.

### 5.2.2.1 Diagram Kelas *Controller*

Pada Gambar 5.4 menggambarkan diagram kelas *controller* yang memiliki 5 subkelas. Kelima subkelas *controller* memiliki hubungan generalisasi terhadap kelas *CI\_Controller*. Secara umum diagram kelas *controller* menyediakan fungsi logika yang merepresentasikan kemampuan sistem untuk memenuhi kebutuhan pengguna. Fungsi logika yang merepresentasikan kemampuan sistem untuk melakukan otentifikasi pengguna ditangani oleh fungsi pada sub kelas *C\_login*.



Gambar 5.4 Diagram Kelas Untuk *Controller*

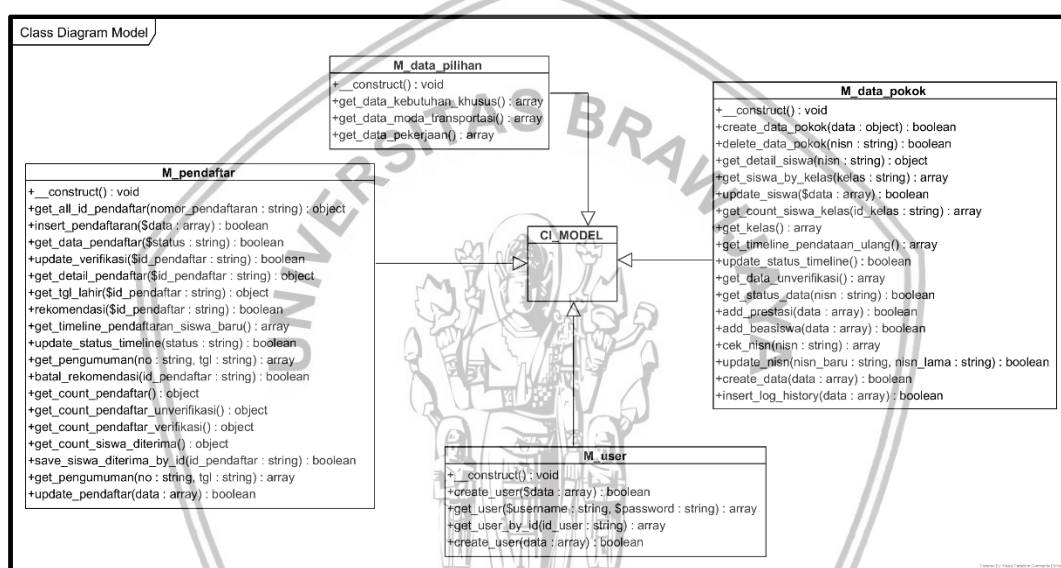
Sedangkan Fungsi logika yang merepresentasikan kemampuan sistem untuk menampilkan formulir pendaftaran dan memasukkan data pendaftaran ditangani oleh fungsi pada subkelas *C\_pendaftaran*. Fungsi pada subkelas *C\_guru* menangani kemampuan sistem untuk mengelola data pendaftaran siswa baru yang dilakukan oleh aktor guru. Selain itu Subkelas *C\_guru* juga memiliki kemampuan untuk menyimpan data siswa yang diterima untuk dijadikan data pokok.

Pada kegiatan pendataan ulang data pokok siswa ditangani oleh subkelas *C\_siswa* dan *C\_petugas\_tata\_usaha*. Subkelas *C\_siswa* memilih fungsi logika

untuk menampilkan formulir pendataan ulang dan juga fungsi logika untuk memperbaharui data pokok tersebut. Sedangkan fungsi logika yang merepresentasikan kemampuan sistem untuk mengelola data pokok siswa ditangani fungsi pada subkelas C\_petugas\_tata\_usaha.

### 5.2.2.2 Diagram Kelas Model

Diagram kelas *model* secara umum menyediakan fungsi untuk melakukan akses data yang tersimpan pada basis data. Identifikasi diagram kelas model didapatkan dari diagram kelas analisis pada Gambar 5.2. Diagram kelas model ditunjukkan pada Gambar 5.5. Diagram kelas *model* memiliki 4 subkelas. Subkelas M\_user memiliki fungsi akses data pada kegiatan otentifikasi pengguna. Sedangkan kelas M\_data\_pilihan memiliki fungsi untuk akses data pilihan pada formulir yaitu data kebutuhan khusus, data moda transportasi dan data pekerjaan.



Gambar 5.5 Diagram Kelas Perancangan Untuk Model

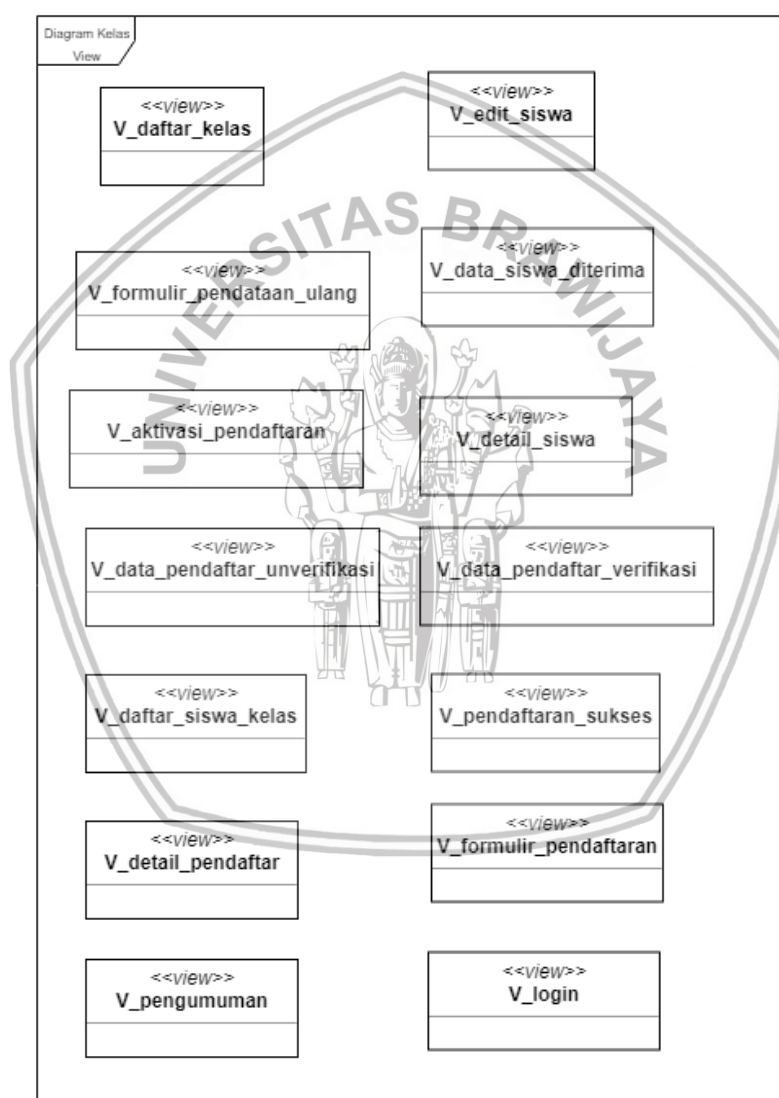
Fungsi akses data pada kegiatan pendaftaran siswa baru ditangani oleh subkelas *model* M\_pendaftar. Salah satu fungsi pada subkelas M\_pendaftar adalah insert\_pendaftaran yang menangani akses data untuk memasukkan data pendaftaran siswa baru. Selain itu terdapat fungsi get\_data\_pendaftar untuk menangani pengambilan semua data pendaftar yang telah tersimpan pada basis data.

Fungsi akses data pada kegiatan pendataan ulang data pokok siswa ditangani oleh subkelas *model* M\_data\_pokok. Salah satu fungsi pada M\_data\_pokok adalah update\_data\_pokok, fungsi ini menangani akses data untuk melakukan pembaharuan data pokok siswa. Selain itu terdapat fungsi get\_detail\_data\_pokok untuk menangani akses pengambilan data pokok siswa yang dipilih.

### 5.2.2.3 Diagram Kelas View

Diagram kelas *view* menyediakan rancangan dalam pembuatan kelas *view* pada *framework CodeIgniter* yang berisikan kode program untuk tampilan

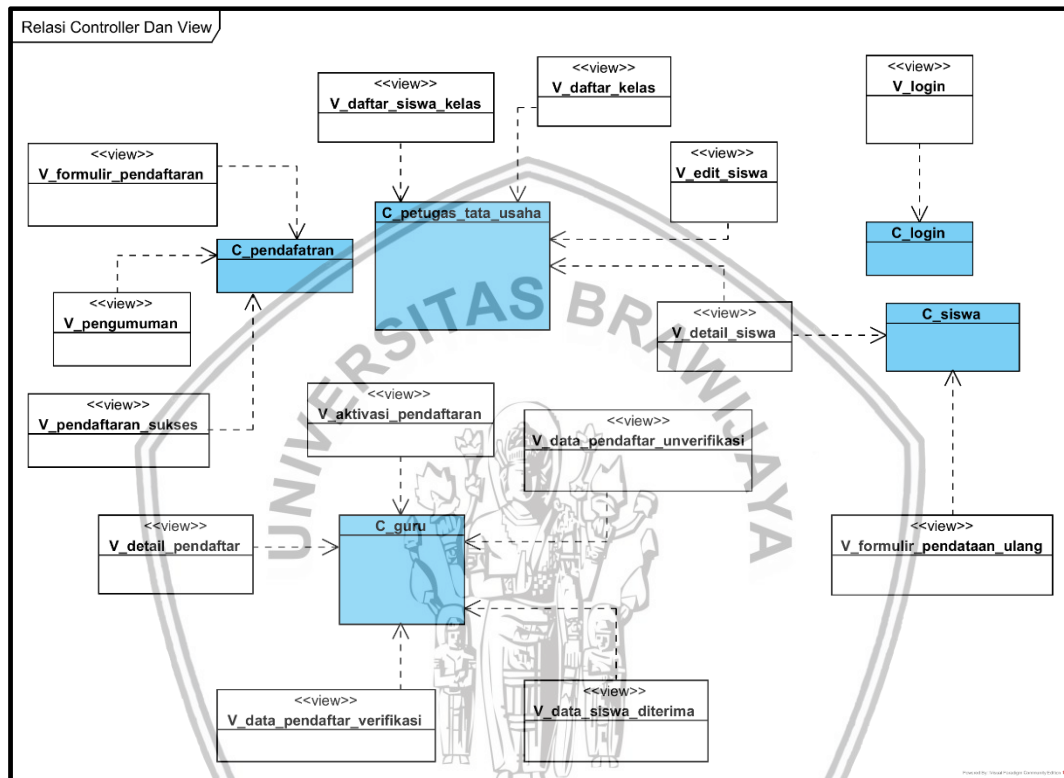
antarmuka pengguna. Diagram kelas view di modelkan dengan kelas yang memiliki *stereotype view* yang akan dijadikan panduan dalam pembuatan dan penamaan file untuk *view*. Pada Gambar 5.6 menggambarkan mengenai Diagram Kelas View yang terdiri 14 kelas. Setiap kelas memiliki hubungan depedensi terhadap file bertipe php yang akan menjadi file untuk tampilan antarmuka pengguna. Diagram kelas *view* memiliki kelas V\_daftar\_kelas, V\_formulir\_pendataan\_ulang, V\_aktivasi\_pendaftaran, V\_data\_pendaftar\_unverifikasi, V\_daftar\_siswa\_kelas, V\_detail\_pendaftar, V\_pengumuman, V\_edit\_siswa, V\_data\_siswa\_diterima, V\_detail\_siswa, V\_data\_pendaftar\_verifikasi, V\_pendaftaran\_sukses, dan V\_formulir\_pendaftaran, V\_login.



**Gambar 5.6 Diagram Kelas Untuk View**

Pada Gambar 5.7 merupakan relasi antara diagram kelas *controller* dan diagram kelas *view*. Setiap kelas *view* memiliki hubungan depedensi terhadap kelas *controller*. Kelas *controller* C\_pendaftar memiliki hubungan depedensi terhadap kelas *view* V\_formulir\_pendaftaran, V\_pengumuman dan

V\_pendaftaran\_sukses. Kelas *controller* C\_petugas\_tata\_usaha memiliki hubungan dengan kelas *view* V\_daftar\_siswa\_kelas, V\_daftar\_kelas, V\_edit\_siswa, dan V\_detail\_siswa. Sedangkan kelas *controller* C\_guru memiliki hubungan dengan kelas *view* V\_aktivasi\_pendaftaran, V\_data\_pendaftar\_unverifikasi, V\_data\_siswa\_diterima, V\_data\_pendaftar\_verifikasi, V\_detail\_pendaftar. Kelas *controller* C\_siswa memiliki hubungan dengan kelas *view* V\_detail\_siswa dan V\_formulir\_pendaftaran\_ulang. Kelas *controller* C\_login memiliki hubungan dengan kelas *view* V\_login.



Gambar 5.7 Relasi Diagram Kelas *View* dan *Controller*

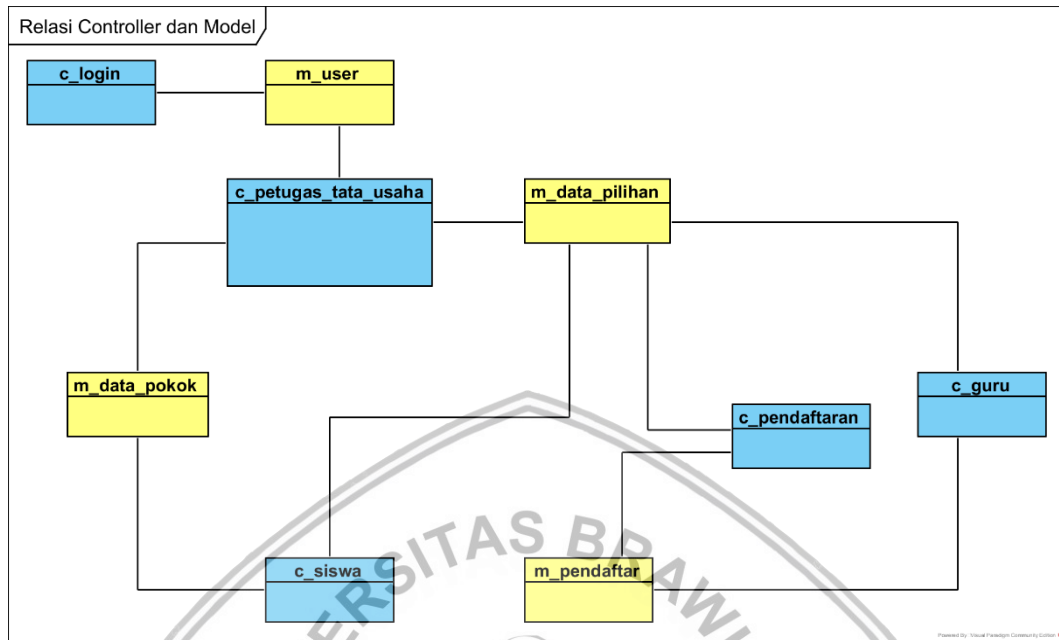
#### 5.2.2.4 Diagram Kelas Perancangan Relasi Subkelas *Controller* dan *Model*

Subbab ini menjelaskan mengenai hubungan dari kelas perancangan *controller* dan kelas perancangan *model*. Gambar 5.8 merupakan visualisasi hubungan antara subkelas *controller* dengan subkelas *model*. Diagram kelas perancangan relasi subkelas *controller* dan *model* akan menjadi dasar dalam tahap implementasi.

Subkelas *controller* digambarkan dengan kelas berwarna biru dan subkelas *model* digambarkan dengan kelas berwarna kuning. Subkelas C\_petugas\_tata\_usaha memiliki hubungan dengan subkelas *model* M\_data\_pokok dan subkelas M\_user. Subkelas C\_pendaftaran memiliki hubungan dengan subkelas *model* M\_pendaftar dan subkelas M\_data\_pilihan. Subkelas C\_guru memiliki hubungan dengan subkelas M\_user dan subkelas M\_pendaftar. Sedangkan subkelas C\_siswa memiliki hubungan dengan subkelas M\_data\_pokok



dan M\_data\_pilihan. Subkelas C\_login memiliki hubungan dengan subkelas M\_user.



Gambar 5.8 Diagram Kelas Relasi Subkelas *Controller* dan *Model*

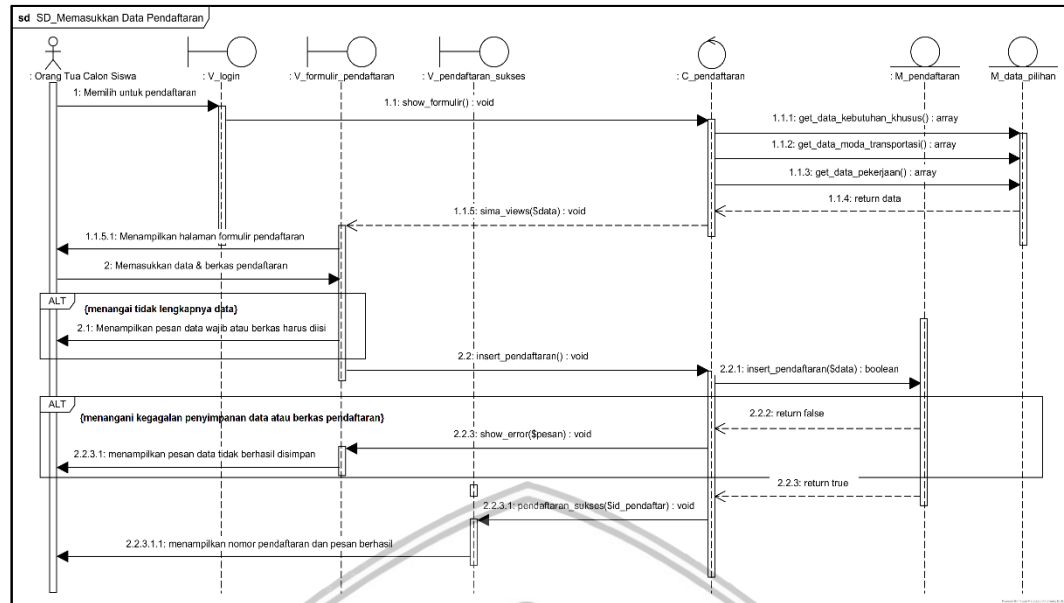
### 5.3 Pemodelan Interaksi Objek Perancangan

Pemodelan interaksi objek dilakukan untuk menggambarkan pertukaran pesan antar entitas aktor, entitas *boundary*, objek *control*, dan objek *model* yang saling berinteraksi untuk memenuhi kebutuhan pengguna. Pemodelan interaksi objek didokumentasikan menggunakan *sequence diagram*. *Sequence diagram* dibuat berdasarkan alur dari beberapa *use case* yang sudah didokumentasikan pada spesifikasi *use case* dan kelas yang telah diidentifikasi pada diagram kelas perancangan.

Beberapa *sequence diagram* yang akan dijelaskan pada bagian ini merupakan visualisasi dari interaksi pada proses memasukkan data pendaftaran siswa baru, verifikasi formulir dan berkas pendaftaran, dan memasukkan data pendataan ulang data pokok siswa.

#### 5.3.1 Memasukkan Data Pendaftaran Siswa Baru

Gambar 5.9 berisi *sequence diagram* yang memvisualisasikan interaksi antar objek yang terlibat dalam proses pendaftaran siswa baru. *Sequence diagram* dimulai ketika aktor Orang Tua Calon siswa mengirimkan pesan untuk memilih pendaftaran melalui objek *boundary* V\_header. Kemudian objek *boundary* meneruskan pesan kepada kelas C\_pendaftaran sebagai objek *control* yang akan mengirimkan pesan kepada objek *model* M\_data\_pilihan untuk mengambil data, kemudian mengirimkannya kepada objek *boundary* V\_formulir\_pendaftaran untuk ditampilkan kepada aktor Orang Tua Calon Siswa.



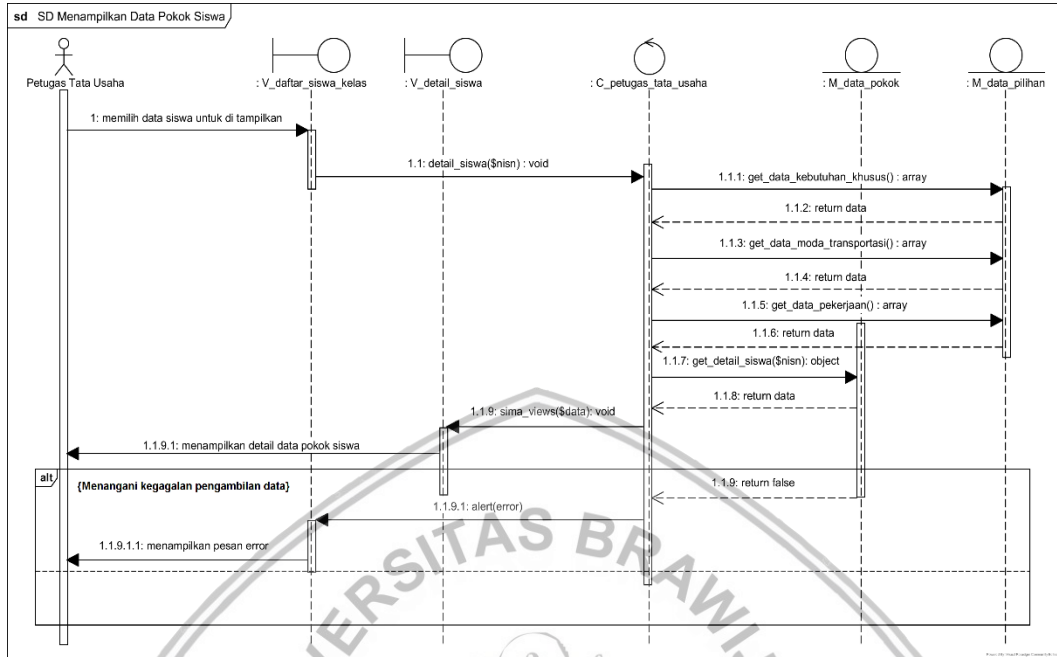
**Gambar 5.9 Sequence Diagram Memasukkan Data Pendaftaran Siswa Baru**

Selanjutnya, aktor Orang Tua Calon Siswa memasukkan data pendaftaran dan juga berkas pendaftaran, objek *boundary* *V\_formulir\_pendaftaran* akan memastikan data yang wajib diisi telah terisi. Jika ada data wajib yang belum diisi maka objek *boundary* akan menampilkan pesan bahwa data wajib harus diisi. Data pendaftaran dan berkas yang telah dimasukkan akan dikirimkan melalui objek *boundary* *V\_formulir\_pendaftaran* kepada objek *controller* *C\_pendaftaran* sebagai objek *control*. Objek *control* akan meneruskan pesan kepada kelas *M\_pendaftaran* sebagai objek *model*. Objek *model* akan menyimpan data pada database. Jika data berhasil disimpan maka objek *model* *M\_pendaftaran* akan mengirimkan pesan *true* kepada objek *controller* *C\_pendaftaran*. kemudian objek *controller* *C\_pendaftaran* akan mengirimkan pesan melalui objek *boundary* *V\_pendaftaran\_sukses* bahwa data berhasil disimpan.

### 5.3.2 Menampilkan Detail Data Pokok Siswa

Gambar 5.10 berisi *sequence diagram* yang memvisualisasikan interaksi antar objek yang terlibat dalam proses mengelola data pokok siswa untuk menampilkan detail data pokok siswa. *Sequence diagram* dimulai ketika aktor petugas tata usaha mengirimkan pesan untuk memilih menampilkan detail data pokok siswa melalui objek *boundary* *V\_daftar\_siswa\_kelas*. Kemudian objek *boundary* meneruskan pesan kepada kelas *C\_petugas\_tata\_usaha* sebagai objek *controller* yang akan mengirimkan pesan kepada objek *model* *M\_data\_pokok* dan *M\_data\_pilihan* untuk mengambil data pokok siswa. kemudian objek *model* *M\_data\_pokok* dan *M\_data\_pilihan* akan mengirimkan data, jika mengalami kegagalan maka objek *model* tersebut akan mengirimkan pesan *false*. Selanjutnya objek *controller* *C\_petugas\_tata\_usaha* akan memanggil objek *boundary*

V\_detail\_siswa dan data yang dihasilkan untuk ditampilkan kepada aktor petugas tata usaha.

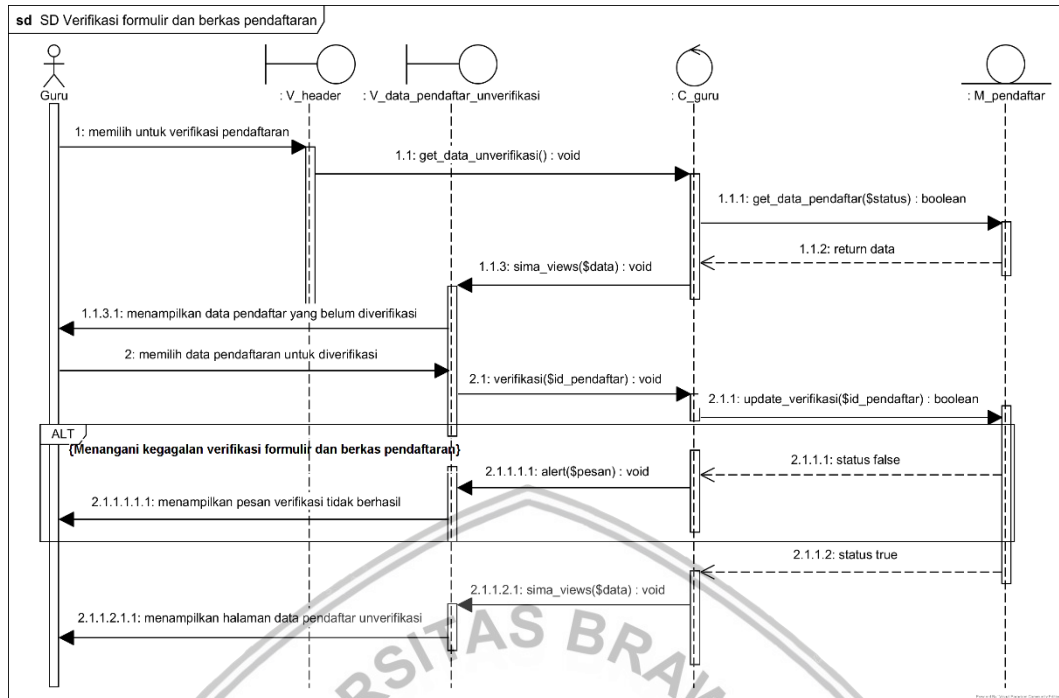


Gambar 5.10 Sequence Diagram Menampilkan Detail Data Pokok Siswa

### 5.3.3 Verifikasi Fomulir dan Berkas Pendaftaran

Sequence diagram yang terdapat pada Gambar 5.11 yang memvisualisasikan interaksi antar objek yang terlibat dalam proses verifikasi formulir dan berkas pendaftaran yang dimulai ketika aktor Guru mengirimkan pesan melalui objek *boundary* V\_header. Kemudian objek *boundary* meneruskan pesan kepada kelas C\_guru sebagai objek *control*. Selanjutnya Objek *control* akan berinteraksi dengan objek model yang ada pada objek *model* M\_pendaftar untuk melakukan proses pemuatan data pendaftar yang belum diverifikasi. Data tersebut akan dikirimkan ke objek *controller* yang kemudian akan diteruskan kepada objek *boundary* V\_data\_pendaftar\_unverifikasi.

Kemudian aktor Guru mengirimkan pesan untuk memverifikasi salah satu data yang telah dipilih melalui objek *boundary* V\_data\_pendaftar\_unverifikasi. Objek *boundary* akan mengirimkan pesan kepada kelas C\_guru sebagai objek *controller*. Objek *controller* meneruskan pesan kepada objek *model* pada kelas M\_pendaftar untuk melakukan perubahan status verifikasi data pendaftar yang dipilih sesuai dengan id\_pendaftar. Kemudian Objek *model* akan mengirimkan pesan sukses kepada objek *Control* pada kelas C\_guru yang akan diteruskan kepada objek *boundary* V\_data\_pendaftar\_unverifikasi.

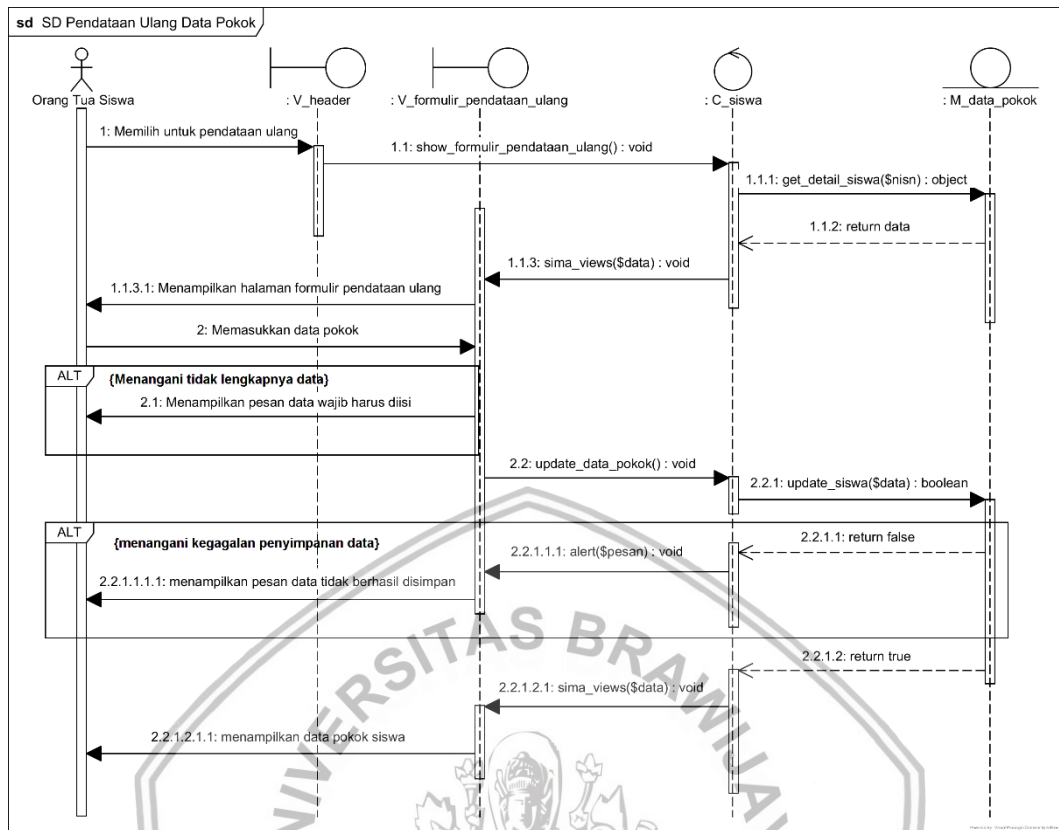


Gambar 5.11 Sequence Diagram Verifikasi Formulir dan Berkas Pendaftaran

### 5.3.4 Memperbaharui Data Pokok Siswa

Gambar 5.12 berisi *sequence diagram* yang memvisualisasikan interaksi antar objek yang terlibat dalam proses pendataan ulang data pokok siswa. *Sequence diagram* dimulai ketika aktor Orang Tua siswa mengirimkan pesan untuk memilih pendataan ulang melalui objek *boundary* V\_header. Kemudian objek *boundary* meneruskan pesan kepada kelas C\_siswa sebagai objek *controller* yang akan melakukan pengambilan data melalui objek *model* M\_data\_pokok, kemudian data dikirimkan kepada objek *boundary* V\_pendataan\_ulang untuk ditampilkan kepada aktor Orang Tua Siswa.

Selanjutnya, aktor Orang Tua Siswa memasukkan data pokok, objek *boundary* V\_pendataan\_ulang akan memastikan data yang wajib diisi telah terisi. Jika ada data wajib yang belum diisi maka objek *boundary* akan menampilkan pesan bahwa data wajib harus diisi. Data pendaftaran dan berkas yang telah dimasukkan akan dikirimkan melalui objek *boundary* V\_formulir\_pendataan\_ulang kepada kelas C\_siswa sebagai objek *control*. Objek *control* akan meneruskan pesan kepada kelas M\_data\_pokok sebagai objek *model*.



**Gambar 5.12 Sequence Diagram Memperbaharui Data Pendataan Ulang Data Pokok Siswa**

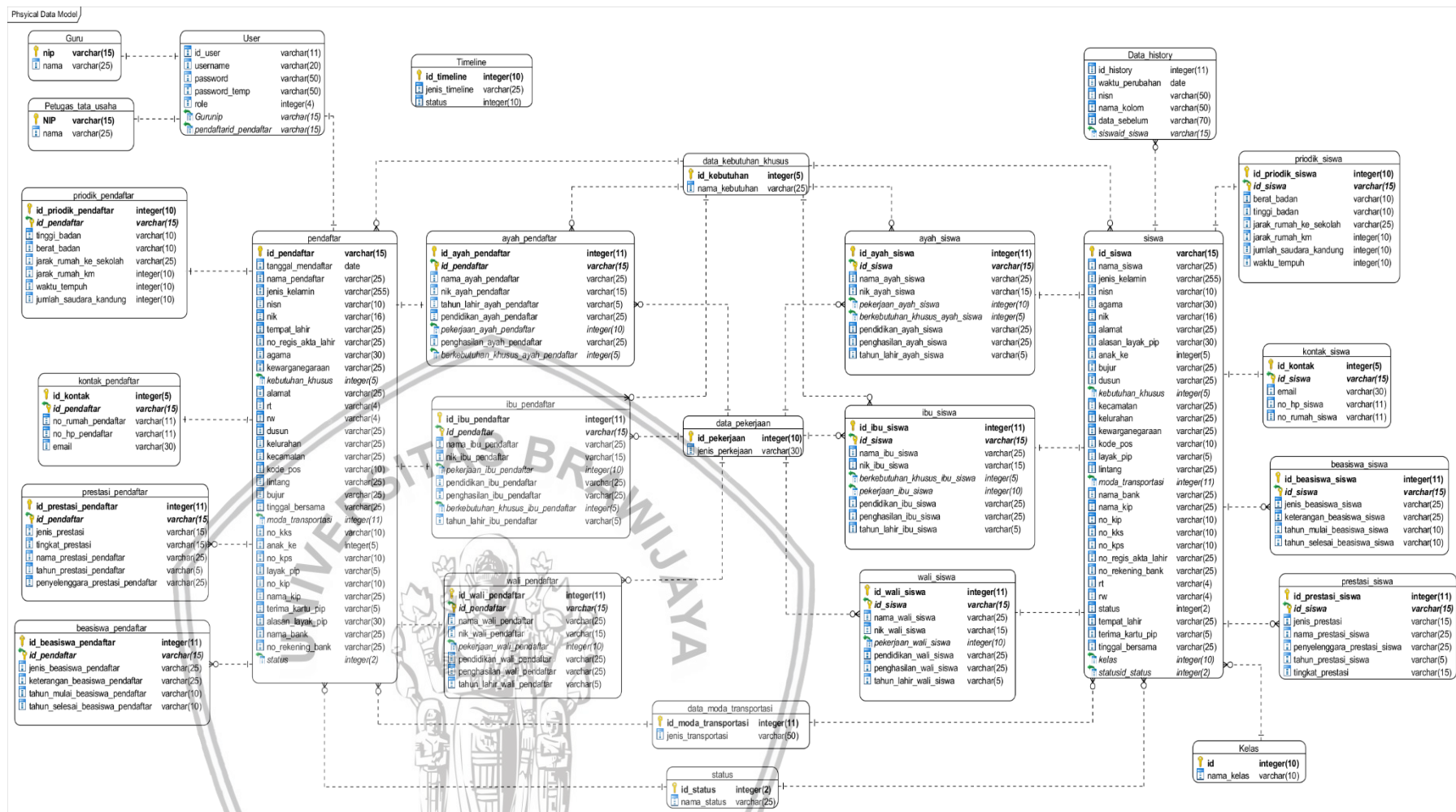
Objek *model* akan menyimpan data pada database. Jika data berhasil disimpan maka objek *model* M\_data\_pokok akan mengirimkan pesan *true* kepada objek *controller* C\_siswa. Kemudian objek *controller* C\_siswa akan mengirimkan pesan melalui objek *boundary* v\_formulir\_pendataan\_ulang untuk menampilkan data pokok siswa yang telah diperbaharui.

## 5.4 Perancangan Basis Data

Rancangan basis data yang terdapat dalam Gambar 5.13 dibuat berdasarkan pemodelan kelas diagram analisis yang ditunjukkan dalam Gambar 5.2. masing-masing kelas pada diagram kelas analisis direpresentasikan oleh tabel yang terdapat pada PDM. Sedangkan atribut yang terdapat di dalam masing-masing kelas analisis direpresentasikan berdasarkan analisis atribut yang ada dalam Gambar 5.3. Hubungan antar tabel pada PDM dibuat berdasarkan hubungan antar kelas pada diagram kelas analisis.

Perancangan basis data terdiri dari 26 tabel. Data modul pendaftaran siswa baru tersimpan di tabel pendaftar yang menyimpan data diri calon siswa, tabel ayah\_pendaftar dan ibu\_pendaftar menyimpan data diri orang tua dari calon siswa, tabel wali\_pendaftar yang menyimpan data diri wali dari calon siswa. Tabel priodik\_pendaftar menyimpan data diri priodik dari calon siswa, tabel kontak\_pendaftar menyimpan data kontak dari calon siswa. Sedangkan tabel





Gambar 5.13 Physical Data Model

prestasi\_pendaftar dan tabel beasiswa\_pendaftar menyimpan data prestasi dan beasiswa yang dimiliki oleh calon siswa. Tabel status menyimpan data jenis status pada kegiatan pendaftaran siswa baru.

Data pada modul pendataan ulang data pokok siswa tersimpan di tabel siswa yang menyimpan data diri setiap siswa, tabel ayah\_siswa dan ibu\_siswa yang menyimpan data diri orang tua dari siswa, tabel wai\_siswa menyimpan data diri wali dari siswa. Tabel priodik\_siswa menyimpan data diri priodik dari siswa, tabel kontak\_siswa menyimpan data kontak dari calon siswa. Sedangkan tabel prestasi siswa dan tabel beasiswa\_siswa menyimpan data prestasi dan beasiswa yang dimiliki oleh siswa. sedangkan Tabel kelas menyimpan data setiap kelas.

Data yang di miliki oleh petugas tata usaha disimpan pada tabel petugas\_tata\_usaha. Data untuk pilihan moda transportasi yang ada pada formulir pendaftaran dan formulir pendataan ulang data pokok siswa disimpan pada tabel\_moda\_transportasi. Tabel data\_pekerjaan menyimpan jenis pekerjaan yang digunakan dalam kegiatan pendaftaran siswa baru dan pendataan ulang data pokok siswa.

Sedangkan tabel data\_kebutuhan\_khusus menyimpan jenis kebutuhan khusus yang digunakan dalam formulir pendaftaran siswa baru dan pendataan ulang data pokok siswa. Hasil perancangan basis data akan digunakan sebagai panduan implementasi basis data yang akan digunakan untuk memenuhi kebutuhan penyimpanan data pada sistem informasi yang akan dikembangkan.

## 5.5 Perancangan Algoritme

Perancangan algoritme merupakan tahap yang dilakukan untuk menentukan dan menyusun sekumpulan operasi logika sebagai konsep pembangunan fungsi-fungsi dari sistem informasi yang akan dikembangkan. Hasil perancangan algoritme pada penelitian ini di dokumentasikan ke dalam beberapa *pseudocode*. Perancangan algoritme yang di dokumentasikan akan menjadi panduan untuk pembangunan sistem informasi dalam tahap implementasi.

### 5.5.1 Menampilkan Formulir Pendaftaran Siswa Baru

Subbab ini berisi rancangan algoritme dari sebuah fungsi yang disediakan oleh sistem kepada pengguna saat pengguna menggunakan sistem untuk menampilkan formulir pendaftaran siswa baru. *Pseudocode* yang terdapat pada Tabel 5.1 merupakan algoritme dari fungsi `show_formulir()` yang terdapat pada kelas `C_pendaftaran`.

**Tabel 5.1 *Pseudocode* Menampilkan Formulir Pendaftaran Siswa Baru**

NO	<i>PSEUDOCODE</i>
1	Start
2	Inisialisasi array data dengan indeks 'code_page' dengan menyimpan string "formulir pendaftaran"
3	Memanggil fungsi untuk menghasilkan nomor pendaftaran dan menyimpannya pada array bernama data dengan indeks 'nomor_pendaftaran'

NO	PSEUDOCODE
4	Memanggil fungsi untuk mengambil data kebutuhan khusus pada basis data dan menyimpannya pada array data dengan indeks 'data_kebutuhan_khusus'
5	Memanggil fungsi untuk mengambil data moda transportasi pada basis data dan menyimpannya pada array data dengan indeks 'data_moda_transportasi'
6	Memanggil fungsi untuk mengambil data pekerjaan pada basis data dan menyimpannya pada array data dengan indeks 'data_pekerjaan'
7	Memanggil fungsi untuk menampilkan view formulir pendaftaran dan mengirimkan array data
8	End

Algoritme dimulai dengan menginisialisasi array data dengan indeks code\_page. Kemudian memanggil fungsi untuk menghasilkan nomor pendaftaran. Selanjutnya algoritme memuat data kebutuhan khusus, data moda transportasi, dan data pekerjaan, semua data tersebut disimpan pada array bernama data. Kemudian menampilkan antarmuka pengguna formulir pendaftaran dengan mengirimkan data.

### 5.5.2 Memasukkan Data Pendaftaran Siswa Baru

Subbab ini berisi rancangan algoritme dari sebuah fungsi yang disediakan oleh sistem kepada pengguna saat pengguna menggunakan sistem untuk menyimpan data pendaftaran siswa baru. *Pseudocode* yang terdapat pada Tabel 5.2 merupakan merupakan algoritme dari fungsi insert\_pendaftaran() yang terdapat pada kelas C\_pendaftaran.

**Tabel 5.2 Pseudocode Memasukkan Data Pendaftaran Siswa Baru**

NO	PSEUDOCODE
1	Start
2	array data dengan indeks 'data_pendaftar' menyimpan data diri pendaftar yang dikirim oleh view formulir pendaftaran
3	array data dengan indeks 'data_ayah' menyimpan data ayah pendaftar yang dikirim oleh view formulir pendaftaran
4	array data dengan indeks 'data_ibu' menyimpan data ibu pendaftar yang dikirim oleh view formulir pendaftaran
5	array data dengan indeks 'data_wali' menyimpan data wali pendaftar yang dikirim oleh view formulir pendaftaran
6	array data dengan indeks 'data_kontak' menyimpan data kontak pendaftar yang dikirim oleh view formulir pendaftaran
7	array data dengan indeks 'data_priodik' menyimpan data priodik pendaftar yang dikirim oleh view formulir pendaftaran
8	array data dengan indeks 'data_prestasi' menyimpan data prestasi pendaftar yang dikirim oleh view formulir pendaftaran
9	array data dengan indeks 'data_beasiswa' menyimpan data beasiswa pendaftar yang dikirim oleh view formulir pendaftaran
10	array data dengan indeks 'data_berkas' menyimpan data ayah pendaftar yang dikirim oleh view formulir pendaftaran

NO	PSEUDOCODE
11	memanggil fungsi insert_pendaftar dengan paramater data
12	jika berhasil maka :
13	tampilkan pesan berhasil
14	jika gagal maka :
15	tampilkan pesan gagal
16	End

Algoritme dimulai dengan menerima data pendaftaran, data ayah pendaftar, data ibu pendaftar, data wali pendaftar, data priodik, data kontak, data prestasi, data beasiswa dan data berkas yang telah dikirimkan dari antarmuka pengguna formulir pendaftaran. Semua data tersebut disimpan pada array data, kemudian semua data tersebut diproses untuk disimpan pada basis data. Jika data berhasil disimpan maka akan menampilkan pesan berhasil, sedangkan jika data gagal disimpan maka akan muncul pesan data tidak berhasil disimpan.

### 5.5.3 Menampilkan Detail Data Pokok Siswa

Subbab ini berisi rancangan algoritme dari sebuah fungsi yang disediakan oleh sistem kepada pengguna saat pengguna menggunakan sistem untuk menampilkan detail data pokok siswa yang diinginkan. *Pseudocode* yang terdapat pada Tabel 5.3 merupakan algoritme dari fungsi detail\_siswa() yang terdapat pada kelas C\_petugas\_tata\_usaha.

**Tabel 5.3 Pseudocode Memasukkan Data Pendaftaran Siswa Baru**

NO	PSEUDOCODE
1	Start
2	Inisialisasi array data dengan indeks 'code_page' dengan menyimpan string "Detail Siswa"
3	Memanggil fungsi untuk mengambil data kebutuhan khusus pada basis data dan menyimpannya pada array data dengan indeks 'data_kebutuhan_khusus'
4	Memanggil fungsi untuk mengambil data moda transportasi pada basis data dan menyimpannya pada array data dengan indeks 'data_moda_transportasi'
5	Memanggil fungsi untuk mengambil data pekerjaan pada basis data dan menyimpannya pada array data dengan indeks 'data_pekerjaan'
6	Memanggil fungsi untuk mengambil data siswa pada basis data dengan parameter nsn dan menyimpannya pada array data dengan indeks data
7	Jika berhasil maka :
8	Memanggil fungsi untuk menampilkan view formulir pendaftaran dan mengirimkan array data
9	Jika gagal maka :
10	menampilkan pesan gagal
11	End

Algoritme dimulai dengan menginisialisasi array data dengan indeks code\_page. Selanjutnya algoritme memuat data kebutuhan khusus, data moda

transportasi, dan data pekerjaan, semua data tersebut disimpan pada array bernama data. Kemudian algoritme memuat data pokok siswa yang ada pada basis data dengan parameter nisn, data pokok disimpan pada array bernama data. Selanjutnya menampilkan antarmuka pengguna detail siswa dengan mengirimkan data.

#### 5.5.4 Memperbaharui Data Pokok Siswa

Subbab ini berisi rancangan algoritme dari sebuah fungsi yang disediakan oleh sistem kepada pengguna saat pengguna menggunakan sistem untuk memperbaharui data pokok siswa. *Pseudocode* yang terdapat pada Tabel 5.4 merupakan rancangan algoritme fungsi `update_data_pokok()` yang terdapat pada kelas `C_siswa`.

Algoritme dimulai dengan menerima data siswa, data ayah siswa, data ibu siswa, data wali siswa, data priodik, data kontak, data prestasi, data beasiswa dan data berkas yang telah dikirimkan dari antarmuka pengguna formulir pendataan ulang. Semua data tersebut disimpan pada array data, kemudian semua data tersebut diproses untuk disimpan pada basis data. Jika data berhasil disimpan maka akan menampilkan pesan berhasil, sedangkan jika data gagal disimpan maka akan muncul pesan data tidak berhasil disimpan.

**Tabel 5.4 Pseudocode Memperbaharui Data Pokok Siswa**

NO	PSEUDOCODE
1	Start
2	array data dengan indeks 'data_siswa' menyimpan data diri pendaftar yang dikirim oleh view formulir pendataan ulang
3	array data dengan indeks 'data_ayah' menyimpan data ayah pendaftar yang dikirim oleh view formulir pendataan ulang
4	array data dengan indeks 'data_ibu' menyimpan data ibu pendaftar yang dikirim oleh view formulir pendataan ulang
5	array data dengan indeks 'data_wali' menyimpan data wali pendaftar yang dikirim oleh view formulir pendataan ulang
6	array data dengan indeks 'data_kontak' menyimpan data kontak pendaftar yang dikirim oleh view formulir pendataan ulang
7	array data dengan indeks 'data_priodik' menyimpan data priodik pendaftar yang dikirim oleh view formulir pendataan ulang
8	array data dengan indeks 'data_prestasi' menyimpan data prestasi pendaftar yang dikirim oleh view formulir pendataan ulang
9	array data dengan indeks 'data_beasiswa' menyimpan data beasiswa pendaftar yang dikirim oleh view formulir pendataan ulang
10	array data dengan indeks 'data_berkas' menyimpan data ayah pendaftar yang dikirim oleh view formulir pendataan ulang
11	memanggil fungsi <code>update_siswa</code> dengan paramater data
12	jika berhasil maka :
13	tampilkan pesan berhasil
14	jika gagal maka :
15	tampilkan pesan gagal



NO	PSEUDOCODE
16	End

## 5.6 Perancangan Antarmuka Pengguna

Subbab perancangan antarmuka pengguna berisi beberapa sketsa tampilan antarmuka pengguna sistem informasi yang akan dikembangkan dalam penelitian ini. Beberapa sketsa antarmuka pengguna sistem di antaranya adalah antarmuka login, pendaftaran siswa, menampilkan data pendaftar, dan formulir pendataan ulang.

### 5.6.1 Antarmuka Pengguna Login

Rancangan antarmuka pengguna *login* ditunjukkan dalam Gambar 5.14 merupakan tampilan yang akan dilihat oleh pengguna pada saat akan melakukan otentifikasi identitas sebelum dapat mengakses informasi melalui sistem berdasarkan hak aksesnya. Antarmuka pengguna *login* merupakan bagian dari *use case login*. Komponen antarmuka *login* di antaranya adalah pada bagian atas terdapat bagian yang berisikan logo sistem, *dropdown* menu 1 dan *dropdown* menu 2. Pada bagian utama terdapat logo sistem, kolom untuk memasukkan nama atau identitas pengguna, kolom untuk memasukkan kata sandi, dan tombol untuk menjalankan fungsi otentifikasi pengguna.

Gambar 5.14 Antarmuka Pengguna Login

### 5.6.2 Antarmuka Pengguna Memasukkan Data Pendaftaran Siswa Baru

Rancangan antarmuka pengguna memasukkan data pendaftaran siswa baru merupakan tampilan sistem yang akan dilihat oleh pengguna, pada saat pengguna menggunakan sistem untuk memasukkan data pendaftaran. Antarmuka pengguna pendaftaran siswa merupakan bagian dari *use case* memasukkan data

pendaftaran siswa baru dan rancangan pseudocode memasukkan data pendaftaran siswa baru.

Antarmuka pengguna pendaftaran siswa yang ditunjukkan pada Gambar 5.15 menunjukkan dua kelompok komponen antarmuka yang berbentuk persegi di sisi kiri dan kanan. Di sisi kiri terdapat *sidebar* yang memiliki beberapa komponen di antaranya logo sistem dan beberapa submenu. Sedangkan di sisi kanan terdapat dua kelompok komponen. Komponen pertama menunjukkan nomor pendaftaran siswa dan komponen tanggal. Sedangkan komponen kedua terdapat *dropdown* yang berisi formulir pendaftaran. Selain itu terdapat juga tombol untuk menjalankan fungsi pendaftaran siswa baru.

LOGO	HEADER	
	<div> <div>NOMOR PENDAFTARAN</div> <div>Tanggal</div> </div>	
MENU 1	<div> <div>DATA SISWA</div> <div>DATA ORANG TUA</div> <div>DATA WALI</div> <div>DATA PRIODIK</div> <div>DATA KONTAK</div> <div>DATA BEASISWA</div> <div>DATA PRESTASI</div> <div>UPLOAD BERKAS</div> <div>SUBMIT</div> </div>	
MENU 2		

Gambar 5.15 Antarmuka Pengguna Pendaftaran Siswa

### 5.6.3 Antarmuka Pengguna Menampilkan Data Pokok Siswa

Rancangan antarmuka Pengguna pengelolaan data pokok siswa yang ditunjukkan pada Gambar 5.16 merupakan tampilan sistem yang akan dilihat oleh pengguna, pada saat pengguna menggunakan sistem untuk menampilkan data pokok siswa. Antarmuka pengguna menampilkan data pokok siswa merupakan bagian dari *use case* mengelola data pokok siswa dan rancangan *pseudocode* menampilkan data pokok siswa.

Pada rancangan antarmuka pengguna menampilkan data pokok siswa terdapat dua kelompok komponen yang berada di dalam persegi di sisi kiri dan

kanan. Di sisi kiri merupakan *sidebar* yang memiliki informasi mengenai pengguna yang memiliki akses, tombol untuk menjalankan fungsi logout dan menampilkan beberapa menu sistem. Sedangkan pada sisi kanan terdapat *header* pada sisi atas, komponen yang menampilkan nsn, nama siswa, kelas, status, dan tombol untuk menjalankan fungsi memperbaharui data pokok siswa, dan tombol untuk menjalankan fungsi cetak data pokok siswa. Kemudian terdapat komponen yang berisikan beberapa *tabmenu*. Setiap *tabmenu* akan menampilkan data pokok siswa.

<b>LOGO</b> <div> <div> NAMA PENGGUNA </div> </div> <div>LOGOUT</div> <div>MENU 1</div> <div>MENU 2</div> <div>MENU 3</div>	<div>HEADER</div> <div> <div>NISN</div> <div>NAMA SISWA</div> <div>KELAS</div> <div>STATUS</div> <div>PRINT DATA</div> <div>EDIT</div> </div> <div> <div>TAB 1</div> <div>TAB 2</div> <div>TAB 3</div> <div>DATA POKOK SISWA</div> </div>
--	---

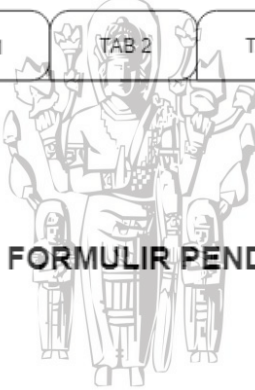
**Gambar 5.16 Antarmuka Pengguna Pengelolaan Data Pendaftar**

#### 5.6.4 Antarmuka Pengguna Memperbaharui Data Pokok Siswa

Rancangan antarmuka pengguna memperbaharui data pokok siswa merupakan tampilan sistem yang akan dilihat oleh siswa, pada saat siswa menggunakan sistem untuk memperbaharui data pokok siswa. Antarmuka pengguna memperbaharui data pokok siswa merupakan bagian dari *use case* mengelola memperbaharui data pokok siswa dan rancangan pseudocode memperbaharui data pokok siswa.

Antarmuka pengguna pendataan ulang data pokok siswa yang ditunjukkan pada Gambar 5.17 menunjukkan dua kelompok komponen antarmuka yang berbentuk persegi di sisi kiri dan kanan.

Di sisi kiri terdapat *sidebar* yang memiliki beberapa komponen di antaranya logo sistem, nama pengguna, tombol untuk menjalankan fungsi *logout* dan beberapa submenu. Sedangkan di sisi kanan terdapat tiga kelompok komponen. Kelompok pertama menunjukkan *header* sistem. Komponen kedua menunjukkan nama siswa, nomor nsn, dan status siswa. Komponen ketiga terdapat *tabmenu* yang berisi formulir pendataan ulang. Selain itu terdapat juga tombol untuk menjalankan fungsi pendataan ulang data pokok siswa.

<b>LOGO</b> <div>○ NAMA PENGGUNA</div> <div>LOGOUT</div> <div>MENU 1</div> <div>MENU 2</div>	<b>HEADER</b>		
	NISN		
	NAMA SISWA		
	STATUS		
	<div>TAB 1    TAB 2    TAB 3</div> <div>  <b>FORMULIR PENDATAAN ULANG</b> </div> <div>SUBMIT</div>		

Gambar 5.17 Antarmuka Pengguna Formulir Pendataan Ulang

## BAB 6 IMPLEMENTASI

Bab implementasi berisi penjelasan mengenai kebutuhan implementasi dan hasil implementasi sistem informasi yang dikembangkan dalam penelitian ini. Bab implementasi menjelaskan mengenai spesifikasi lingkungan implementasi, beberapa kode program, dan beberapa gambar antarmuka pengguna sistem informasi yang dikembangkan. Implementasi dilakukan berdasarkan hasil perancangan yang telah dijelaskan pada bab sebelumnya.

### 6.1 Spesifikasi Lingkungan Implementasi

Subbab spesifikasi lingkungan implementasi menjelaskan spesifikasi perangkat keras dan perangkat lunak yang digunakan selama proses pembangunan sistem informasi. Perangkat keras yang digunakan pada proses pembangunan sistem informasi adalah sebuah laptop dengan spesifikasi yang dijelaskan pada Tabel 6.1. Perangkat lunak yang digunakan untuk pengembangan ditunjukkan pada Tabel 6.2. Sedangkan spesifikasi lingkungan *deployment* ditunjukkan pada Tabel 6.3.

**Tabel 6.1 Spesifikasi Perangkat Keras**

<b>Unit Komputasi</b>	Laptop ASUS A455L
<b>CPU</b>	Core i5-5200U 2.19 Ghz
<b>Kapasitas RAM</b>	8 Gb
<b>Kapasitas Penyimpanan</b>	500 Gb
<b>Kartu Grafis</b>	NVidia GeForce 930M
<b>Resolusi Layar</b>	1366 x 768 pixels

**Tabel 6.2 Spesifikasi Perangkat Lunak**

<b>Sistem Operasi</b>	Windows 10 Pro 64-bit
<b>Web Server</b>	Apache 2.4.23
<b>DBMS</b>	MariaDB 10.1.16
<b>Bahasa Pemrograman</b>	PHP 7.0.9
<b>Editor Kode Program</b>	Atom
<b>Peramban</b>	Google Chrome Version 70.0.3538.110 (Official Build) (64-bit)
<b>Perangkat Lunak atau basa Pemrograman Pendukung</b>	Javascript
	HTML

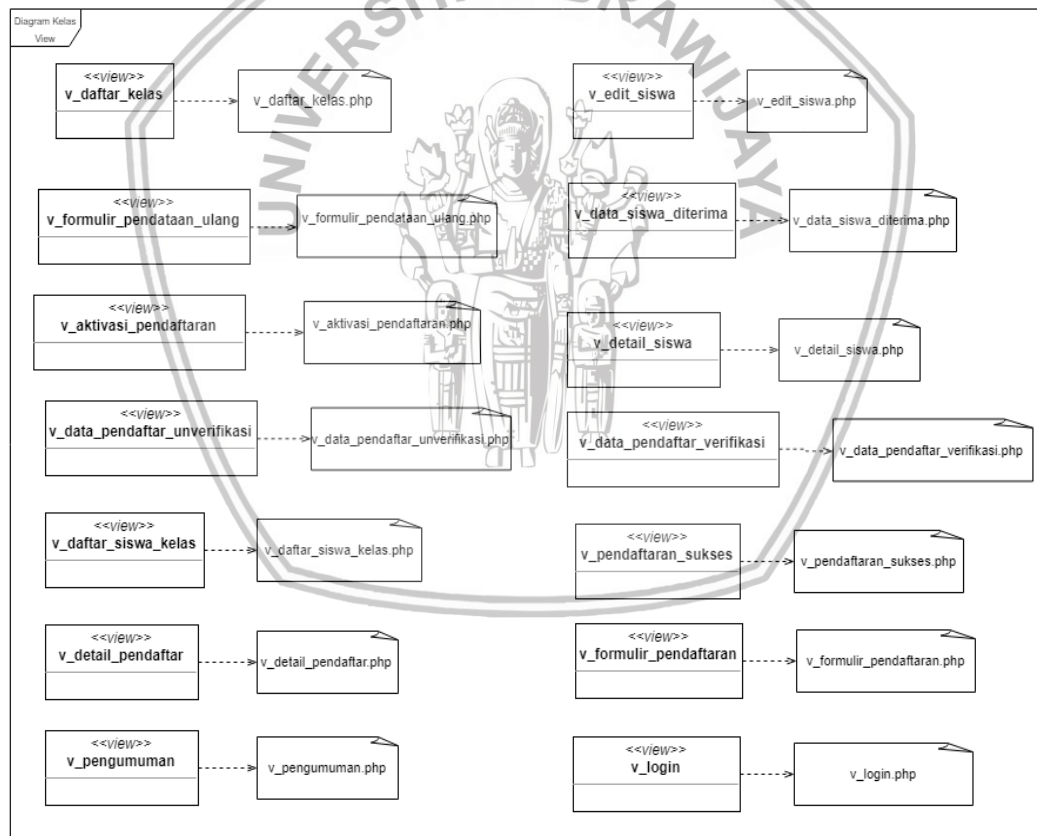


**Tabel 6.3 Spesifikasi Minimal Lingkungan *Deployment***

Kapasitas RAM (Minimal)	256 MB
Kapasitas Memori (Minimal)	85 MB
Versi PHP	5.3 atau lebih
<i>Library</i> Pendukung	FDPF versi 1.6

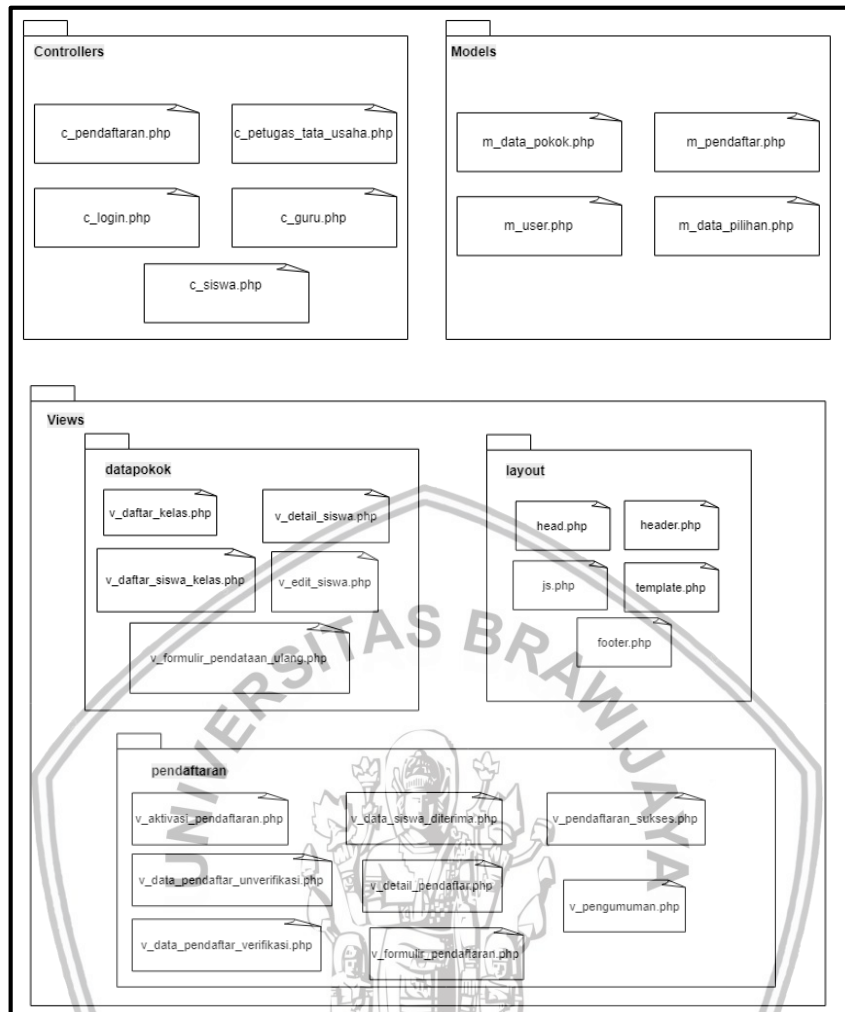
## 6.2 Struktur Artefak Sistem

Struktur artefak sistem memberikan informasi mengenai artefak sistem dalam sistem informasi yang dikembangkan. Informasi mengenai struktur artefak sistem didapatkan dari implementasi diagram kelas *view* yang digambarkan pada Gambar 6.1. Pada Gambar 6.1 menggambarkan implementasi diagram kelas *view* yang dimodelkan dengan hubungan *dependensi* terhadap file bertipe *php*.



**Gambar 6.1 Implementasi Diagram Kelas untuk *View***

Struktur artefak sistem pada Gambar 6.2 menunjukkan terdapat 3 *package* yang terdiri dari *package* controllers, models, dan views. *Package* controllers berisikan implementasi dari diagram kelas controller yang terdiri dari file c\_pendaftaran.php, c\_petugas\_tata\_usaha.php, c\_login.php, c\_guru.php, c\_siswa.php.



**Gambar 6.2 Struktur Artefak Sistem**

*Package* models berisikan implementasi dari diagram kelas model yang terdiri dari file `m_data_pokok.php`, `m_pendaftar`, `m_user`, dan `m_data_pilihan`. Sedangkan *package* views berisikan implementasi dari antarmuka pengguna. *Package* views terdapat 3 *package* yang terdiri dari *package* `datapokok`, *package* `layout`, dan *package* `pendaftaran`. Pada *package* `datapokok` berisikan file `v_daftar_kelas.php`, `v_detail_siswa.php`, `v_daftar_siswa_kelas.php`, `v_edit_siswa.php`, `v_formulir_pendaftaran_ulang.php`. Pada *package* `layout` berisikan file `head.php`, `header.php`, `js.php`, `template.php`, dan `footer.php`. Sedangkan untuk *package* `pendaftaran` terdapat file `v_aktivasi_pendaftaran.php`, `v_data_siswa_diterima.php`, `v_pendaftaran_sukses.php`, `v_data_pendaftar_unverifikasi.php`, `v_detail_pendaftar.php`, `v_pengumuman.php`, `v_data_pendaftar_verifikasi.php`, dan `v_formulir_pendaftaran.php`.

### 6.3 Implementasi Algoritme

Subbab implementasi algoritme berisi beberapa kode program yang dibuat berdasarkan algoritme dalam *pseudocode* yang telah dibuat pada tahap

perancangan. Beberapa hasil implementasi algoritme terdiri dari kode program menampilkan formulir pendaftaran siswa baru, memasukkan data pendaftaran siswa baru, mengelola data pokok siswa, dan memperbaharui data pokok siswa.

### 6.3.1 Menampilkan Formulir Pendaftaran Siswa Baru

Kode program menampilkan formulir pendaftaran siswa baru merupakan bagian dari *use case* memasukkan data pendaftaran siswa baru yang disediakan oleh sistem. Implementasi algoritme menampilkan formulir pendaftaran siswa baru dibuat berdasarkan hasil rancangan algoritme menampilkan formulir pendaftaran siswa baru. Tabel 6.4 merupakan penerapan algoritme dari fungsi `show_formulir()` yang terdapat pada kelas `C_pendaftaran`.

**Tabel 6.4 Implementasi Algoritme Menampilkan Formulir Pendaftaran Siswa Baru**

No	Kode Program
1	<code>\$data['code_page'] = "Formulir Pendaftaran";</code>
2	<code>    \$data['nomor_pendaftaran'] = \$this-&gt;</code>
3	<code>&gt;generate_nomor_pendaftaran();</code>
4	<code>    \$data['data_kebutuhan_khusus'] = \$this-&gt;m_data_pilihan-&gt;</code>
5	<code>&gt;get_data_kebutuhan_khusus();</code>
6	<code>    \$data['data_moda_transportasi'] = \$this-&gt;</code>
7	<code>&gt;m_data_pilihan-&gt;get_data_moda_transportasi();</code>
8	<code>    \$data['data_pekerjaan'] = \$this-&gt;m_data_pilihan-&gt;</code>
9	<code>&gt;get_data_pekerjaan();</code>
10	<code>    \$data['timeline'] = \$this-&gt;m_pendaftar-&gt;</code>
11	<code>&gt;get_timeline_pendaftaran_siswa_baru();</code>
12	
13	<code>    \$this-&gt;template-&gt;</code>
14	<code>&gt;sima_views('pendaftaran/v formulir pendaftaran',\$data);</code>

### 6.3.2 Memasukkan Data Pendaftaran Siswa Baru

Kode program memasukkan data pendaftaran siswa baru merupakan bagian dari *use case* memasukkan data pendaftaran siswa baru yang disediakan oleh sistem. Implementasi algoritme memasukkan data pendaftaran siswa baru dibuat berdasarkan hasil rancangan algoritme memasukkan data pendaftaran siswa baru. Tabel 6.5 merupakan merupakan algoritme dari fungsi `insert_pendaftaran()` yang terdapat pada kelas `C_pendaftaran`.

**Tabel 6.5 Implementasi Algoritme Memasukkan Data Pendaftaran Siswa Baru**

No	Kode Program
1	<code>\$data['data_pendaftar'] = array(</code>

No	Kode Program
2	"id_pendaftar" => \$this->input->post("nomor_pendaftaran"),
3	"nama_pendaftar" => \$this->input->post("nama_pendaftar"),
4	"nisn" => \$this->input->post("nisn_pendaftar"),
5	"jenis_kelamin" => \$this->input->post("radio_jenis_kelamin"),
6	"kewarganegaraan" => \$this->input->post("radio_kewarganegaraan"),
7	"tempat_lahir" => \$this->input->post("tempat_lahir_pendaftar"),
8	"tanggal_lahir" => \$this->input->post("tanggal_lahir_pendaftar"),
9	"alamat" => \$this->input->post("alamat_rumah_pendaftar"),
10	"agama" => \$this->input->post("radio_agama"),
11	"tinggal_bersama" => \$this->input->post("tinggal_bersama_pendaftar"),
12	"moda_transportasi" => \$this->input->post("transportasi_pendaftar"),
13	"rt" => \$this->input->post("rt_pendaftar"),
14	"rw" => \$this->input->post("rw_pendaftar"),
15	"lintang" => \$this->input->post("lintang_pendaftar"),
16	"bujur" => \$this->input->post("bujur_pendaftar"),
17	"kelurahan" => \$this->input->post("kelurahan_pendaftar"),
18	"dusun" => \$this->input->post("dusun_pendaftar"),
19	"kecamatan" => \$this->input->post("kecamatan_pendaftar"),
20	"kode_pos" => \$this->input->post("kode_pos_pendaftar"),
21	"kebutuhan_khusus" => \$this->input->post("kebutuhan_pendaftar"),
22	"nik" => \$this->input->post("nik_pendaftar"),
23	"no_regis_akta_lahir" => \$this->input->post("no_registrasi_akta_lahir"),
24	"alasan_layak_pip" => \$this->input->post("alasan_layak_pip"),
25	"no_kks" => \$this->input->post("no_kks"),
26	"no_kps" => \$this->input->post("no_kps_pendaftar"),
27	"no_kip" => \$this->input->post("no_kip_pendaftar"),
28	"nama_kip" => \$this->input->post("nama_kip"),
29	"nama_bank" => \$this->input->post("nama_bank_pendaftar"),
30	"no_rekening_bank" => \$this->input->post("no_rekening_bank"),
31	"nama_pada_rekening" => \$this->input->post("nama_rekening"),
32	"terima_kartu_kip" => \$this->input->post("radio_terima_kartu_fisik_kip"),

No	Kode Program
33	"layak_pip" => \$this->input->post("radio_usulan_pip"),
34	"anak_ke" => \$this->input->post("anak_ke"),
35	"status" => 1,
36	"tanggal_mendaftar" => date('y-m-d H:i:s')
37	);
38	\$data['data_ayah'] = array(
39	"id_pendaftar" => \$this->input->post("nomor_pendaftaran"),
40	"nama_ayah_pendaftar" => \$this->input->post("nama_ayah_pendaftar"),
41	"nik_ayah_pendaftar" => \$this->input->post("nik_ayah_pendaftar"),
42	"tahun_lahir_ayah_pendaftar" => \$this->input->post("tahun_lahir_ayah_pendaftar"),
43	"pendidikan_ayah_pendaftar" => \$this->input->post("pendidikan_ayah_pendaftar"),
44	"pekerjaan_ayah_pendaftar" => \$this->input->post("pekerjaan_ayah_pendaftar"),
45	"penghasilan_ayah_pendaftar" => \$this->input->post("penghasilan_ayah_pendaftar"),
46	"berkebutuhan_khusus_ayah_pendaftar" => \$this->input->post("kebutuhan_ayah_pendaftar"),
47	);
48	\$data['data_ibu'] = array(
49	"id_pendaftar" => \$this->input->post("nomor_pendaftaran"),
50	"nama_ibu_pendaftar" => \$this->input->post("nama_ibu_pendaftar"),
51	"nik_ibu_pendaftar" => \$this->input->post("nik_ibu_pendaftar"),
52	"tahun_lahir_ibu_pendaftar" => \$this->input->post("tahun_lahir_ibu_pendaftar"),
53	"pendidikan_ibu_pendaftar" => \$this->input->post("pendidikan_ibu_pendaftar"),
54	"pekerjaan_ibu_pendaftar" => \$this->input->post("pekerjaan_ibu_pendaftar"),
55	"penghasilan_ibu_pendaftar" => \$this->input->post("penghasilan_ibu_pendaftar"),
56	"berkebutuhan_khusus_ibu_pendaftar" => \$this->input->post("kebutuhan_ibu_pendaftar"),
57	);
58	\$data['data_wali'] = array(
59	"id_pendaftar" => \$this->input->post("nomor_pendaftaran"),
60	"nama_wali_pendaftar" => \$this->input->post("nama_wali_pendaftar"),
61	"nik_wali_pendaftar" => \$this->input->post("nik_wali_pendaftar"),
62	"tahun_lahir_wali_pendaftar" => \$this->input->post("tahun_lahir_wali_pendaftar"),
63	"pendidikan_wali_pendaftar" => \$this->input->post("pendidikan_wali_pendaftar"),



No	Kode Program
64	"pekerjaan_wali_pendaftar" => \$this->input-
65	>post("pekerjaan_wali_pendaftar"),
66	"penghasilan_wali_pendaftar" => \$this->input-
67	>post("penghasilan_wali_pendaftar")
68	);
69	\$data['data_kontak'] = array(
70	"id_pendaftar" => \$this->input-
71	>post("nomor_pendaftaran"),
72	"no_rumah_pendaftar" => \$this->input-
73	>post("no_rumah_pendaftar"),
74	"no_hp_pendaftar" => \$this->input-
75	>post("no_hp_pendaftar"),
76	"email" => \$this->input->post("email_pendaftar")
77	);
78	\$data['data_priodik'] = array(
79	"id_pendaftar" => \$this->input-
80	>post("nomor_pendaftaran"),
81	"tinggi_badan" => \$this->input-
82	>post("tinggi_pendaftar"),
83	"berat_badan" => \$this->input-
84	>post("berat_pendaftar"),
85	"jumlah_saudara_kandung" => \$this->input-
86	>post("jumlah_saudara_pendaftar"),
87	"jarak_rumah_ke_sekolah" => \$this->input-
88	>post("radio_jarak_rumah"),
89	"jarak_rumah_km" => \$this->input-
90	>post("jarak_pendaftar"),
91	"waktu_tempuh" => \$this->input-
92	>post("waktu_tempuh_pendaftar")
93	);
94	if(\$this->input-
95	>post("jenis_prestasi_pendaftar_3") != "PILIHAN"){
96	\$data['count_prestasi'] = 4;
97	}elseif (\$this->input-
98	>post("jenis_prestasi_pendaftar_2") != "PILIHAN") {
99	\$data['count_prestasi'] = 3;
100	}elseif (\$this->input-
101	>post("jenis_prestasi_pendaftar_1") != "PILIHAN") {
102	\$data['count_prestasi'] = 2;
103	}else{
104	\$data['count_prestasi'] = 1;
105	}
106	for (\$i=1; \$i < \$data['count_prestasi'] ; \$i++) {
107	\$data['prestasi'][\$i] = array(
108	"id_pendaftar" => \$this->input-
109	>post("nomor_pendaftaran"),
110	"jenis_prestasi" => \$this->input-
111	>post("jenis_prestasi_pendaftar_\$i"),
112	"tingkat_prestasi" => \$this->input-
113	>post("tingkat_prestasi_pendaftar_\$i"),
114	"nama_prestasi_pendaftar" => \$this->input-
115	>post("nama_prestasi_pendaftar_\$i"),

No	Kode Program
97	"tahun_prestasi_pendaftar" => \$this->input-
98	>post("tahun_prestasi_pendaftar_{\$i}"),
99	"penyelenggara_prestasi_pendaftar" => \$this-
100	>input->post("penyelenggara_prestasi_pendaftar_{\$i}"),
101	);
102	}
103	if(\$this->input-
104	>post("jenis_beasiswa_pendaftar_3") != "PILIHAN"){
105	\$data['count_beasiswa'] = 4;
106	}elseif (\$this->input-
107	>post("jenis_beasiswa_pendaftar_2") != "PILIHAN") {
108	\$data['count_beasiswa'] = 3;
109	}elseif (\$this->input-
110	>post("jenis_beasiswa_pendaftar_1") != "PILIHAN") {
111	\$data['count_beasiswa'] = 2;
112	}else{
113	\$data['count_beasiswa'] = 1;
114	}
115	for (\$i=1; \$i < \$data['count_beasiswa'] ; \$i++) {
116	\$data['beasiswa'][\$i] = array(
117	"id_pendaftar" => \$this->input-
118	>post("nomor_pendaftar"),
119	"jenis_beasiswa_pendaftar" => \$this->input-
120	>post("jenis_beasiswa_pendaftar_{\$i}"),
121	"keterangan_beasiswa_pendaftar" => \$this-
122	>input->post("nama_beasiswa_pendaftar_{\$i}"),
123	"tahun_mulai_beasiswa_pendaftar" => \$this-
124	>input->post("tahun_mulai_beasiswa_pendaftar_{\$i}"),
125	"tahun_selesai_beasiswa_pendaftar" => \$this-
126	>input->post("tahun_selesai_beasiswa_pendaftar_{\$i}")
127	);
128	}
129	//upload berkas
130	\$config['allowed_types'] = 'gif jpg png'; //
131	jenis file
132	\$config['max_size'] = 3000;
133	\$config['max_width'] = 1024;
134	\$config['max_height'] = 768;
135	\$config['upload_path'] =
136	'./public/berkas/file_akta_kelahiran/';
137	// upload berkas akta kelahiran
138	\$config['file_name'] = 'akta-kelahiran-
139	.'. \$this->input->post("nomor_pendaftar");
140	\$this->upload->initialize(\$config);
141	\$this->upload->do_upload('AKTA_KELAHIRAN');
142	\$data['data_pendaftar']['file_akta_kelahiran'] =
143	\$this->upload->data('file_name');
144	// upload berkas ijazah tk
145	\$config['upload_path'] =
146	'./public/berkas/file_ijazah tk/';

No	Kode Program
132	<code>\$config['file_name'] = 'ijazah-tk-'. \$this-&gt;input-&gt;post("nomor_pendaftaran");</code>
133	<code>\$this-&gt;upload-&gt;initialize(\$config);</code>
134	<code>\$this-&gt;upload-&gt;do_upload('IJAZAH_TK');</code>
135	<code>\$data['data_pendaftar']['file_ijazah_tk'] = \$this-&gt;upload-&gt;data('file_name');</code>
136	
137	<code>\$sukses = \$this-&gt;m_pendaftar-&gt;insert_pendaftaran(\$data);</code>
138	<code>if(\$sukses){</code>
139	<code>    \$this-&gt;pendaftaran_sukses(\$this-&gt;input-&gt;post("nomor_pendaftaran"));</code>
140	<code>}else{</code>
141	<code>    \$this-&gt;show_error("Data Pendaftaran Tidak Berhasil Dimasukkan");</code>
142	<code>}</code>

### 6.3.3 Menampilkan Detail Data Pokok Siswa

Kode program menampilkan detail data pokok siswa yang merupakan bagian dari *use case* mengelola data pokok siswa yang disediakan oleh sistem. Implementasi algoritme menampilkan detail data pokok siswa dibuat berdasarkan hasil rancangan algoritme menampilkan detail data pokok siswa. Tabel 6.6 merupakan algoritme dari fungsi detail\_siswa() yang terdapat pada kelas C\_petugas\_tata\_usaha.

**Tabel 6.6 Implementasi Algoritme Menampilkan Detail Data Pokok Siswa**

No	Kode Program
1	<code>\$data['code_page'] = "Detail Siswa";</code>
2	<code>\$data['data_kebutuhan_khusus'] = \$this-&gt;m_data_pilihan-&gt;get_data_kebutuhan_khusus();</code>
	<code>\$data['data_moda_transportasi'] = \$this-&gt;m_data_pilihan-&gt;get_data_moda_transportasi();</code>
3	<code>\$data['data_pekerjaan'] = \$this-&gt;m_data_pilihan-&gt;get_data_pekerjaan();</code>
4	<code>\$data['data'] = \$this-&gt;m_data_pokok-&gt;get_detail_siswa(\$nisn);</code>
5	<code>if(\$data['data']['data'] != null){</code>
6	<code>    \$this-&gt;template-&gt;sima_views('datapokok/v_detail_siswa',\$data);</code>
7	<code>}else{</code>
8	<code>echo "&lt;script type='text/javascript'&gt;alert('Nomor Ndisn atau Pengambilan Data salah');&lt;/script&gt;";</code>
9	<code>redirect('petugas-tata-usaha');</code>
10	<code>}</code>
11	

### 6.3.4 Memperbaharui Data Pokok Siswa

Kode program memperbaharui data pokok siswa yang merupakan bagian dari *use case* memperbaharui data pokok siswa yang disediakan oleh sistem. Implementasi algoritme menampilkan detail data pokok siswa dibuat berdasarkan

hasil rancangan algoritme memperbaharui data pokok siswa. Tabel 6.7 merupakan rancangan algoritme fungsi update\_data\_pokok() yang terdapat pada kelas C\_siswa.

**Tabel 6.7 Implementasi Algoritme Memperbaharui Data Pokok Siswa**

No	Kode Program
1	\$data['data_siswa'] = array(
2	"nama_siswa" => \$this->input->post("nama_siswa"),
3	"nisn" => \$this->input->post("nisn_siswa"),
4	"jenis_kelamin" => \$this->input->
5	>post("radio_jenis_kelamin"),
6	"kewarganegaraan" => \$this->input->
7	>post("radio_kewarganegaraan"),
8	"tempat_lahir" => \$this->input->
9	>post("tempat_lahir_siswa"),
10	"tanggal_lahir" => \$this->input->
11	>post("tanggal_lahir_siswa"),
12	"alamat" => \$this->input->post("alamat_rumah_siswa"),
13	"agama" => \$this->input->post("radio_agama"),
14	"tinggal_bersama" => \$this->input->
15	>post("tinggal_bersama_siswa"),
16	"moda_transportasi" => \$this->input->
17	>post("transportasi_siswa"),
18	"rt" => \$this->input->post("rt_siswa"),
19	"rw" => \$this->input->post("rw_siswa"),
20	"lintang" => \$this->input->post("lintang_siswa"),
21	"bujur" => \$this->input->post("bujur_siswa"),
22	"kelurahan" => \$this->input->post("kelurahan_siswa"),
23	"dusun" => \$this->input->post("dusun_siswa"),
24	"kecamatan" => \$this->input->post("kecamatan_siswa"),
25	"kode_pos" => \$this->input->post("kode_pos_siswa"),
26	"kebutuhan_khusus" => \$this->input->
27	>post("kebutuhan_siswa"),
28	"nik" => \$this->input->post("nik_siswa"),
29	"no_regis_akta_lahir"=> \$this->input->
30	>post("no_registrasi_akta_lahir"),
31	"alasan_layak_pip"=> \$this->input->
32	>post("alasan_layak_pip"),
	"no_kks" => \$this->input->post("no_kks"),
	"no_kps" => \$this->input->post("no_kps_siswa"),
	"no_kip" => \$this->input->post("no_kip_siswa"),
	"nama_kip" => \$this->input->post("nama_kip"),
	"nama_bank" => \$this->input->post("nama_bank_siswa"),
	"no_rekening_bank" => \$this->input->
	>post("no_rekening_bank"),
	"nama_pada_rekening" => \$this->input->
	>post("nama_rekening"),
	"terima_kartu_kip" => \$this->input->
	>post("radio_terima_kartu_fisik_kip"),
	"layak_pip" => \$this->input->
	>post("radio_usulan_pip"),

No	Kode Program
33	"anak_ke" => \$this->input->post("anak_ke"),
34	"status_data" => 0
35	);
36	
37	\$data['data_ayah'] = array(
38	"id_siswa" => \$this->input->post("nisn_siswa"),
39	"nama_ayah_siswa" => \$this->input-
40	>post("nama_ayah_siswa"),
41	"nik_ayah_siswa" => \$this->input-
42	>post("nik_ayah_siswa"),
43	"tahun_lahir_ayah_siswa" => \$this->input-
44	>post("tahun_lahir_ayah_siswa"),
45	"pendidikan_ayah_siswa" => \$this->input-
46	>post("pendidikan_ayah_siswa"),
47	"pekerjaan_ayah_siswa" => \$this->input-
48	>post("pekerjaan_ayah_siswa"),
49	"penghasilan_ayah_siswa" => \$this->input-
50	>post("penghasilan_ayah_siswa"),
51	"berkebutuhan_khusus_ayah_siswa" => \$this->input-
52	>post("kebutuhan_ayah_siswa"),
53	);
54	
55	\$data['data_ibu'] = array(
56	"id_siswa" => \$this->input->post("nisn_siswa"),
57	"nama_ibu_siswa" => \$this->input-
58	>post("nama_ibu_siswa"),
59	"nik_ibu_siswa" => \$this->input-
60	>post("nik_ibu_siswa"),
61	"tahun_lahir_ibu_siswa" => \$this->input-
62	>post("tahun_lahir_ibu_siswa"),
63	"pendidikan_ibu_siswa" => \$this->input-
64	>post("pendidikan_ibu_siswa"),
65	"pekerjaan_ibu_siswa" => \$this->input-
66	>post("pekerjaan_ibu_siswa"),
67	"penghasilan_ibu_siswa" => \$this->input-
68	>post("penghasilan_ibu_siswa"),
69	"berkebutuhan_khusus_ibu_siswa" => \$this->input-
70	>post("kebutuhan_ibu_siswa"),
71	);
72	
73	\$data['data_wali'] = array(
74	"id_siswa" => \$this->input->post("nisn_siswa"),
75	"nama_wali_siswa" => \$this->input-
76	>post("nama_wali_siswa"),
77	"nik_wali_siswa" => \$this->input-
78	>post("nik_wali_siswa"),
79	"tahun_lahir_wali_siswa" => \$this->input-
80	>post("tahun_lahir_wali_siswa"),
81	"pendidikan_wali_siswa" => \$this->input-
82	>post("pendidikan_wali_siswa"),
83	"pekerjaan_wali_siswa" => \$this->input-
84	>post("pekerjaan_wali_siswa"),



No	Kode Program
66	<code>"penghasilan_wali_siswa" =&gt; \$this-&gt;input-&gt;post("penghasilan_wali_siswa")</code>
67	<code>);</code>
68	
69	<code>\$data['data_kontak'] = array(</code>
70	<code>    "id_siswa" =&gt; \$this-&gt;input-&gt;post("nisn_siswa"),</code>
71	<code>    "no_rumah_siswa" =&gt; \$this-&gt;input-&gt;post("no_rumah_siswa"),</code>
72	<code>    "no_hp_siswa" =&gt; \$this-&gt;input-&gt;post("no_hp_siswa"),</code>
73	<code>    "email" =&gt; \$this-&gt;input-&gt;post("email_siswa")</code>
74	<code>);</code>
75	
76	<code>\$data['data_priodik'] = array(</code>
77	<code>    "id_siswa" =&gt; \$this-&gt;input-&gt;post("nisn_siswa"),</code>
78	<code>    "tinggi_badan" =&gt; \$this-&gt;input-&gt;post("tinggi_siswa"),</code>
79	<code>    "berat_badan" =&gt; \$this-&gt;input-&gt;post("berat_siswa"),</code>
80	<code>    "jumlah_saudara_kandung" =&gt; \$this-&gt;input-&gt;post("jumlah_saudara_siswa"),</code>
81	<code>    "jarak_rumah_ke_sekolah" =&gt; \$this-&gt;input-&gt;post("radio_jarak_rumah"),</code>
82	<code>    "jarak_rumah_km" =&gt; \$this-&gt;input-&gt;post("jarak_siswa"),</code>
83	<code>    "waktu_tempuh" =&gt; \$this-&gt;input-&gt;post("waktu_tempuh_siswa")</code>
84	<code>);</code>
85	
86	<code>if(\$this-&gt;input-&gt;post("jenis_prestasi_siswa_3") != "PILIHAN") {</code>
87	<code>    \$data['count_prestasi'] = 4;</code>
88	<code>}elseif (\$this-&gt;input-&gt;post("jenis_prestasi_siswa_2") != "PILIHAN") {</code>
89	<code>    \$data['count_prestasi'] = 3;</code>
90	<code>}elseif (\$this-&gt;input-&gt;post("jenis_prestasi_siswa_1") != "PILIHAN") {</code>
91	<code>    \$data['count_prestasi'] = 2;</code>
92	<code>}else{</code>
93	<code>    \$data['count_prestasi'] = 1;</code>
94	<code>}</code>
95	
96	<code>for (\$i=1; \$i &lt; \$data['count_prestasi'] ; \$i++) {</code>
97	<code>    \$data['prestasi'][\$i] = array(</code>
98	<code>        "id_prestasi_siswa" =&gt; \$this-&gt;input-&gt;post("id_prestasi_siswa_\$i"),</code>
99	<code>        "id_siswa" =&gt; \$this-&gt;input-&gt;post("nisn_siswa"),</code>
100	<code>        "jenis_prestasi" =&gt; \$this-&gt;input-&gt;post("jenis_prestasi_siswa_\$i"),</code>
101	<code>        "tingkat_prestasi" =&gt; \$this-&gt;input-&gt;post("tingkat_prestasi_siswa_\$i"),</code>

No	Kode Program
102	<pre> "nama_prestasi_siswa" =&gt; \$this-&gt;input- &gt;post("nama_prestasi_siswa_{\$i}"), 103 "tahun_prestasi_siswa" =&gt; \$this-&gt;input- &gt;post("tahun_prestasi_siswa_{\$i}"), 104 "penyelenggara_prestasi_siswa" =&gt; \$this-&gt;input- &gt;post("penyelenggara_prestasi_siswa_{\$i}"), 105 ); 106 } 107 108 if(\$this-&gt;input-&gt;post("jenis_beasiswa_siswa_3") != "PILIHAN") { 109 \$data['count_beasiswa'] = 4; 110 }elseif (\$this-&gt;input-&gt;post("jenis_beasiswa_siswa_2") != "PILIHAN") { 111 \$data['count_beasiswa'] = 3; 112 }elseif (\$this-&gt;input-&gt;post("jenis_beasiswa_siswa_1") != "PILIHAN") { 113 \$data['count_beasiswa'] = 2; 114 }else{ 115 \$data['count_beasiswa'] = 1; 116 } 117 118 for (\$i=1; \$i &lt; \$data['count_beasiswa'] ; \$i++) { 119 \$data['beasiswa'][\$i] = array( 120 "id_beasiswa_siswa" =&gt; \$this-&gt;input- &gt;post("id_beasiswa_siswa_{\$i}"), 121 "id_siswa" =&gt; \$this-&gt;input-&gt;post("nisn_siswa"), 122 "jenis_beasiswa_siswa" =&gt; \$this-&gt;input- &gt;post("jenis_beasiswa_siswa_{\$i}"), 123 "keterangan_beasiswa_siswa" =&gt; \$this-&gt;input- &gt;post("nama_beasiswa_siswa_{\$i}"), 124 "tahun_mulai_beasiswa_siswa" =&gt; \$this-&gt;input- &gt;post("tahun_mulai_beasiswa_siswa_{\$i}"), 125 "tahun_selesai_beasiswa_siswa" =&gt; \$this-&gt;input- &gt;post("tahun_selesai_beasiswa_siswa_{\$i}") 126 ); 127 } 128 129 \$data_sebelum = \$this-&gt;m_data_pokok- &gt;get_detail_siswa(\$this-&gt;input-&gt;post("nisn_siswa")); 130 \$data_sesudah = array_merge(\$data['data_siswa'], \$data['data_ayah'], \$data['data_ibu'], \$data['data_wali'], \$data['data_kontak'], \$data['data_priodik']); 131 132 \$sukses = \$this-&gt;m_data_pokok-&gt;update_siswa(\$data); 133 if(\$sukses) { 134 \$this-&gt;saveLog(\$data_sebelum, \$data_sesudah); 135 redirect(base_url('siswa')); 136 }else{ </pre>

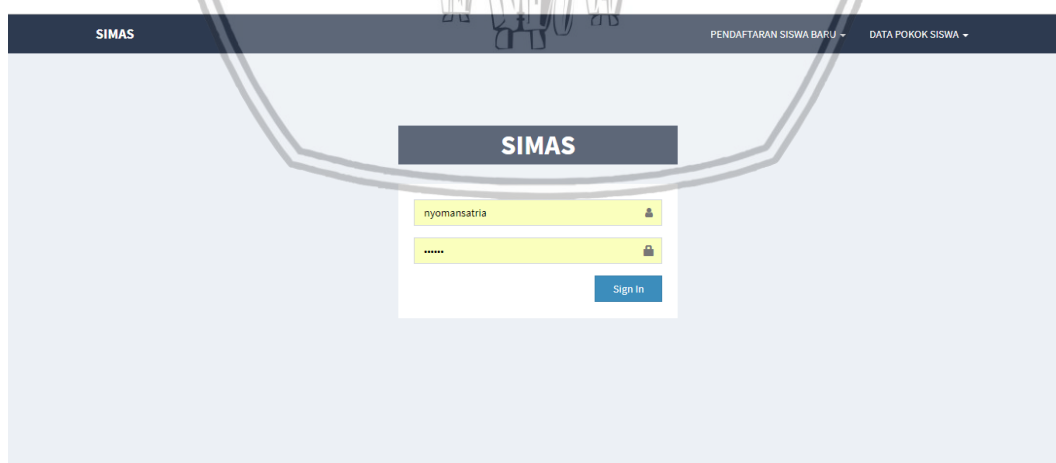
No	Kode Program
137	<code>\$this-&gt;show_error("Data Tidak Berhasil</code>
138	<code>Dimasukkan");</code> <code>}</code>

## 6.4 Implementasi Antarmuka Pengguna

Subbab implementasi antarmuka pengguna menjelaskan beberapa hasil implementasi dalam potongan gambar antarmuka pengguna sistem informasi. Subbab berisi beberapa hasil implementasi antarmuka pengguna diantaranya antarmuka pengguna *login*, antarmuka pengguna, antarmuka pengguna pendaftaran siswa baru, antarmuka pengguna menampilkan data pokok siswa, antarmuka pengguna memperbaharui data pokok siswa. Implementasi antarmuka pengguna dilakukan berdasarkan hasil perancangan antarmuka pengguna yang telah dilakukan dalam penelitian ini.

### 6.4.1 Antarmuka Pengguna Login

Implementasi antarmuka pengguna *login* ditampilkan kepada pengguna melalui peramban pada laptop atau komputer pada saat menggunakan sistem untuk melakukan otentikasi identitas pengguna. Implementasi Antarmuka pengguna *login* merupakan penerapan dari rancangan antarmuka pengguna login. Antarmuka pengguna pada Gambar 6.3 menunjukkan beberapa komponen antarmuka di antaranya judul atau logo sistem informasi, kolom untuk memasukkan identitas pengguna dan memasukkan kata sandi pengguna, dan tombol untuk menjalankan fungsi otentikasi identitas pengguna sistem. Kemudian pada bagian atas terdapat 2 *dropdown* menu untuk menjalankan fungsi pendaftaran siswa baru dan mengelola data pokok siswa.



Gambar 6.3 Antarmuka Pengguna Login

### 6.4.2 Antarmuka Pengguna Memasukkan Data Pendaftaran Siswa Baru

Antarmuka pengguna memasukkan data pendaftaran siswa baru merupakan tampilan sistem yang akan dilihat oleh pendaftar, pada saat pendaftar menggunakan sistem untuk memasukkan data pendaftaran. Antarmuka pengguna

memasukkan data pendaftaran siswa baru merupakan penerapan rancangan antarmuka pengguna memasukkan data pendaftaran siswa baru. Pada Gambar 6.4 menunjukkan beberapa komponen antarmuka pengguna memasukkan data pendaftaran siswa baru.

Hasil implementasi antarmuka pengguna pendaftaran siswa menunjukkan komponen *sidebar* yang berisi nama sistem informasi di bagian atas, dan beberapa menu sistem. Pada bagian utama halaman terdapat dua bagian, bagian pertama menunjukkan nomor pendaftaran siswa dan juga tanggal. Bagian kedua berisi *dropdown* yang berisi formulir pendaftaran siswa baru. Setiap *dropdown* mengelompokkan kolom untuk memasukkan data diri, yang terdiri dari data pribadi, data orang tua, data wali, kontak, data priodik, prestasi, beasiswa, dan data berkas. Selain itu terdapat juga tombol untuk menjalankan fungsi pendaftaran siswa baru.

**Gambar 6.4 Antarmuka Pengguna Memasukkan Data Pendaftaran Siswa Baru**

#### 6.4.3 Antarmuka Pengguna Menampilkan Data Pokok Siswa

Antarmuka Pengguna menampilkan data pokok siswa merupakan penerapan dari rancangan antarmuka pengguna menampilkan data pokok siswa yang akan ditampilkan kepada petugas tata usaha untuk menampilkan data pokok siswa. Antarmuka pengguna menampilkan data pokok siswa ditunjukkan pada Gambar 6.5 menunjukkan beberapa komponen antarmuka pengguna.

Terdapat komponen *sidebar* yang menampilkan nama sistem informasi, informasi mengenai pengguna yang menggunakan sistem, tombol *logout*, dan beberapa menu sistem. Pada bagian utama halaman terdapat 2 bagian, bagian pertama menampilkan kolom untuk nisan, nama siswa, kelas, status siswa dan tombol untuk menjalankan fungsi memperbaharui data siswa dan tombol untuk menjalankan fungsi mencetak data. Sedangkan bagian kedua terdapat *tabmenu* yang berisikan data pokok siswa, *tabmenu* tersebut terdiri dari *tabmenu* data pribadi, data orang tua, data wali, kontak, data priodik, data prestasi, data beasiswa, dan data berkas.

Gambar 6.5 Antarmuka Pengguna Menampilkan Data Pokok Siswa

#### 6.4.4 Antarmuka Pengguna Memperbaharui Data Pokok Siswa

Antarmuka Pengguna memperbaharui data pokok siswa merupakan penerapan dari rancangan antarmuka pengguna memperbaharui data pokok siswa yang akan ditampilkan kepada siswa untuk memperbaharui data pokok siswa. Antarmuka pengguna memperbaharui data pokok siswa ditunjukkan pada Gambar 6.6 menunjukkan beberapa komponen antarmuka pengguna.

Terdapat komponen *sidebar* yang menampilkan nama sistem informasi, informasi mengenai pengguna yang menggunakan sistem, tombol *logout*, dan beberapa menu sistem. Pada bagian utama halaman terdapat 2 bagian, bagian pertama menampilkan kolom informasi dan kolom untuk menampilkan nisn, nama siswa, dan status. Sedangkan bagian kedua terdapat *tabmenu* yang berisikan data pokok siswa, *tabmenu* tersebut terdiri dari *tabmenu* data pribadi, data orang tua, data wali, kontak, data priodik, data prestasi, data beasiswa, dan data berkas.

Gambar 6.6 Antarmuka Pengguna Memperbaharui Data Pokok Siswa



## BAB 7 PENGUJIAN

Pada bab pengujian menjelaskan hasil tahap pengujian yang dilakukan pada sistem informasi manajemen akademik. Beberapa pengujian yang dilakukan terdiri dari pengujian validasi dan pengujian kompatibilitas peramban.

### 7.1 Pengujian Validasi

Pengujian validasi dilakukan untuk memastikan bahwa sistem dapat berjalan sesuai dengan persyaratan yang telah diidentifikasi dan ekspektasi pengguna. Subbab ini berisi hasil pengujian validasi terhadap beberapa fungsi yang digunakan oleh pengguna untuk memasukkan data pendaftaran, memperbaharui data pokok siswa, dan mengelola data pokok siswa.



Gambar 7.1 Kodifikasi Pengujian Validasi

Pengujian validasi dilakukan berdasarkan skenario yang didapatkan dari beberapa alur *use case* memasukkan data pendaftaran siswa baru, mengelola data pokok siswa, dan memperbaharui data pokok siswa. Kemudian kasus uji didapatkan berdasarkan skenario uji yang ditentukan. Pada Gambar 7.1 merupakan kodifikasi pengujian validasi sebagai identitas dalam pengujian validasi.

#### 7.1.1 Pengujian Validasi Memasukkan Data Pendaftaran Siswa Baru

Pengujian validasi memasukkan data pendaftaran siswa baru menjelaskan pengujian beberapa fungsi yang dapat digunakan pengguna untuk melakukan pendaftaran siswa baru. Beberapa skenario pada alur *use case* memasukkan data pendaftaran siswa baru dijelaskan pada Tabel 7.1 yang digunakan untuk mengidentifikasi kasus uji.

Tabel 7.1 Matriks Skenario Pengujian Validasi Memasukkan Data Pendaftaran Siswa Baru

Nomor Skenario	Alur Awal	Alur Alternatif
Skenario 1	<i>Basic Flow</i>	-
Skenario 2	<i>Basic Flow</i>	A1. Menangani tidak lengkapnya data

#### 7.1.1.1 Pengujian Validasi Fungsi Memasukkan Data Pendaftaran Siswa Baru

Tabel 7.2 berisi rencana pengujian dan kasus uji validasi untuk memastikan bahwa sistem dapat digunakan menyimpan data pendaftaran siswa baru dengan kode persyaratan FR-SIAS-11. Kasus uji didapatkan berdasarkan Skenario 1 pada Tabel 7.1 Pengujian validasi fungsi memasukkan data pendaftaran siswa baru menunjukkan hasil valid.

**Tabel 7.2 Rencana Pengujian dan Kasus Uji Fungsi Memasukkan Data Pendaftaran Siswa Baru**

<b>Kode Pengujian</b>	VT-SIAS-01
<b>Nomor Skenario</b>	Skenario 1
<b>Kode Persyaratan</b>	FR-SIAS-11
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk memastikan bahwa sistem dapat digunakan untuk memasukkan data pendaftaran siswa baru dan dapat menyimpan semua data pendaftaran.
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Penguji teridentifikasi sebagai Orang Tua Calon Siswa.</li> <li>2. Penguji memilih fungsi menampilkan formulir pendaftaran siswa baru</li> <li>3. Penguji memasukkan semua data pendaftaran siswa baru</li> <li>4. Penguji memilih fungsi menyimpan data pendaftaran siswa baru.</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem berhasil menyimpan semua data pendaftaran dan menampilkan halaman pendaftaran sukses.
<b>Hasil Pengujian</b>	Sistem berhasil menyimpan semua data pendaftaran dan menampilkan halaman pendaftaran sukses.
<b>Status Pengujian</b>	Valid

#### 7.1.1.2 Pengujian Validasi Fungsi Menangani Tidak Lengkapnya Data

Tabel 7.3 berisi rencana pengujian dan kasus uji validasi untuk memastikan bahwa sistem dapat digunakan menampilkan pesan peringatan jika data wajib belum terisi, dengan kode persyaratan FR-SIAS-08. Kasus uji didapatkan berdasarkan Skenario 2 pada Tabel 7.1 Pengujian validasi fungsi memasukkan data pendaftaran siswa baru menunjukkan hasil valid.

**Tabel 7.3 Rencana Pengujian dan Kasus Uji Fungsi Menangani Tidak Lengkapnya Data**

<b>Kode Pengujian</b>	VT-SIAS-02
-----------------------	------------

<b>Nomor Skenario</b>	Skenario 2
<b>Kode Persyaratan</b>	FR-SIAS-08
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk memastikan bahwa sistem dapat memberikan pesan peringatan jika data wajib belum terisi.
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Penguji teridentifikasi sebagai Orang Tua Calon siswa.</li> <li>2. Penguji memilih fungsi menampilkan formulir pendaftaran siswa baru</li> <li>3. Penguji mengisi semua data pada formulir pendaftaran siswa baru kecuali data nama lengkap.</li> <li>4. Penguji fungsi menyimpan data pendaftaran siswa baru.</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem menampilkan pesan kolom nama lengkap harus diisi dan sistem tidak menyimpan data pendaftaran.
<b>Hasil Pengujian</b>	Sistem menampilkan pesan kolom nama lengkap harus diisi dan sistem tidak menyimpan data pendaftaran.
<b>Status Pengujian</b>	Valid

### 7.1.2 Pengujian Validasi Mengelola Data Pokok Siswa

Pengujian validasi mengelola data pokok siswa menjelaskan pengujian beberapa fungsi yang dapat digunakan pengguna untuk mengelola data pokok siswa. Beberapa skenario pada alur *use case* mengelola data pokok siswa dijelaskan pada Tabel 7.4 yang digunakan untuk mengidentifikasi kasus uji.

**Tabel 7.4 Matriks Skenario Pengujian Validasi Mengelola Data Pokok Siswa**

<b>Nomor Skenario</b>	<b>Alur Awal</b>	<b>Alur Alternatif</b>
Skenario 1	<i>Basic Flow</i>	-
Skenario 2	<i>Basic Flow</i>	S2. Merubah Data Pokok Siswa

#### 7.1.2.1 Pengujian Validasi Fungsi Menampilkan Detail Data Pokok Siswa

Tabel 7.5 berisi rencana pengujian dan kasus uji validasi untuk memastikan bahwa sistem dapat digunakan menampilkan detail data pokok siswa yang dipilih dengan kode persyaratan FR-SIAS-35. Kasus uji didapatkan berdasarkan Skenario 1 pada Tabel 7.4. Pengujian validasi fungsi menampilkan detail data pokok siswa menunjukkan hasil valid.

**Tabel 7.5 Rencana Pengujian dan Kasus Uji Fungsi Menampilkan Detail Data Pokok Siswa**

<b>Kode Pengujian</b>	VT-SIAS-03
<b>Nomor Skenario</b>	Skenario 1
<b>Kode Persyaratan</b>	FR-SIAS-35
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk memastikan bahwa sistem dapat digunakan menampilkan secara detail data pokok siswa.
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Penguji teridentifikasi sebagai petugas tata usaha</li> <li>2. Penguji memilih satu daftar kelas untuk ditampilkan.</li> <li>3. Penguji memilih satu data pokok untuk ditampilkan secara lengkap.</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem berhasil menampilkan secara lengkap data pokok siswa yang dipilih.
<b>Hasil Pengujian</b>	Sistem berhasil menampilkan secara lengkap data pokok siswa yang dipilih.
<b>Status Pengujian</b>	Valid

#### 7.1.2.2 Pengujian Validasi Fungsi Merubah Data Pokok Siswa





Tabel 7.6 berisi rencana pengujian dan kasus uji validasi untuk memastikan bahwa sistem dapat digunakan merubah data pokok siswa, dengan kode persyaratan FR-SIAS-36. Kasus uji didapatkan berdasarkan Skenario 2 pada Tabel 7.4 Pengujian validasi fungsi merubah data pokok siswa hasil valid.

**Tabel 7.6 Rencana Pengujian dan Kasus Uji Fungsi Merubah Data Pokok Siswa**

<b>Kode Pengujian</b>	VT-SIAS-04
<b>Nomor Skenario</b>	Skenario 2
<b>Kode Persyaratan</b>	FR-SIAS-36
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk memastikan bahwa sistem dapat menyimpan perubahan data pokok siswa.
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Penguji teridentifikasi sebagai petugas tata usaha</li> <li>2. Penguji memilih satu daftar kelas untuk ditampilkan</li> <li>3. Penguji memilih satu data pokok untuk ditampilkan secara lengkap.</li> <li>4. Penguji memilih fungsi untuk merubah data.</li> <li>5. Penguji melakukan perubahan data.</li> <li>6. Penguji memilih fungsi untuk menyimpan perubahan data.</li> </ol>



























<b>Hasil yang diharapkan</b>	Sistem berhasil menyimpan data pokok yang telah diperbaharui dan menampilkan halaman detail data pokok.
<b>Hasil Pengujian</b>	Sistem berhasil menyimpan data pokok yang telah diperbaharui dan menampilkan halaman detail data pokok.
<b>Status Pengujian</b>	Valid

## 7.2 Pengujian Kompatibilitas Peramban

-  Tidak ada kendala pada peramban
-  Critical Issue: Terdapat konten atau fungsi yang hilang pada beberapa peramban
-  Major Issues: Terdapat masalah utama pada konten atau performa pada beberapa peramban
-  Minor Issues: Terdapat masalah minor pada konten atau performa pada beberapa peramban

**Gambar 7.2 Kategori Kompatibilitas Pada Aplikasi SortSite**

Subbab ini menjelaskan hasil pengujian kompatibilitas sistem informasi yang menunjukkan kesesuaian sistem dengan beberapa aplikasi peramban menggunakan aplikasi SortSite. Gambar 7.2 menjelaskan mengenai kategori skor untuk mengukur tingkat kompatibilitas pada aplikasi SortSite.

Browser	IE	Edge	Firefox	Safari	Opera	Chrome	iOS			Android	
Version	11	17	63	12	55	70	≤ 10	11	12	≤ 3	4*
Critical Issues											
Major Issues											
Minor Issues											

**Gambar 7.3 Hasil Pengujian Kompatibilitas Peramban**

Pengujian kompatibilitas peramban dilakukan pada peramban yang diakses oleh pengguna yang diidentifikasi sebagai petugas tata usaha. Gambar 7.3 merupakan hasil pengujian kompatibilitas peramban yang menunjukkan bahwa terdapat *Critical Issues* pada jenis peramban firefox versi 63 dan Chrome versi 70, hal ini terjadi karena tidak aktifnya fitur https saat melakukan pengujian. Kemudian terdapat *Major issues* apabila situs diakses melalui peramban Internet Explorer versi 11 dan Andriod versi 4 atau sebelumnya. *Majos issues* terjadi karena kedua peramban tersebut tidak mendukung html dengan tag *input* bertipe *date*.



## BAB 8 PENUTUP

### 8.1 Kesimpulan

Kesimpulan yang didapat berdasarkan penelitian yang dilakukan bahwa Sistem Informasi Manajemen Akademik pada SDN Dinoyo 1 dapat dikembangkan dengan melakukan beberapa kegiatan pengembangan sistem sebagai berikut:

1. Hasil analisis persyaratan berupa dokumentasi yang berisi 2 hasil pemodelan proses bisnis saat ini, analisis masalah berdasarkan proses bisnis saat ini, 2 hasil pemodelan proses bisnis saat ini, identifikasi tipe pemangku kepentingan, kebutuhan pemangku kepentingan, dan identifikasi pengguna. Kemudian identifikasi fitur untuk mengetahui persyaratan fungsional dan persyaratan non fungsional dari sistem yang dikembangkan. Hasil analisis persyaratan juga memuat 9 fitur, 44 persyaratan fungsional dan 1 persyaratan nonfungsional yang merepresentasikan kemampuan sistem secara umum. Hasil analisis persyaratan juga memuat informasi mengenai tujuan penggunaan sistem dan urutan aktivitas yang dilakukan untuk mencapai tujuan tersebut. Informasi tersebut di dokumentasikan pada 7 *use case*. Setiap *use case* akan dijelaskan urutan aktivitas utama dan aktivitas alternatif yang didokumentasikan pada spesifikasi *use case*.
2. Hasil perancangan sistem yang dilakukan berdasarkan hasil analisis persyaratan terdiri dari rancangan arsitektur sistem untuk mengetahui struktur komponen dan hubungan antar komponen pada kerangka kerja *CodeIgniter* dengan menerapkan pola *Model-View-Controller*. Hasil perancangan juga memuat pemodelan objek yang terdiri dari diagram kelas analisis, analisis atribut, diagram kelas perancangan untuk *controller*, diagram kelas perancangan untuk *model*, dan diagram kelas perancangan untuk *view*. Diagram kelas analisis terdiri dari 26 kelas, dan di setiap kelas memiliki atribut-atribut yang didokumentasikan pada analisis atribut. Diagram kelas analisis menjadi panduan dalam perancangan basis data yang memiliki 26 tabel di dalamnya. Kemudian hasil perancangan juga menghasilkan Diagram kelas perancangan untuk *controller* menghasilkan 5 kelas, diagram kelas perancangan untuk *model* menghasilkan 4 kelas, dan diagram kelas untuk *view* menghasilkan 14 kelas. Hasil perancangan sistem juga menghasilkan beberapa pemodelan interaksi objek yang divisualisasikan ke dalam *sequence diagram*, *Physical Data Model* yang berisi tabel-tabel sebagai media penyimpanan data sistem informasi, beberapa rancangan algoritme sistem, dan beberapa rancangan antarmuka pengguna.
3. Hasil implementasi berdasarkan hasil perancangan sistem berupa Sistem Informasi Manajemen Akademik (SIMA) berbasis situs web dengan kerangka kerja *CodeIgniter* dengan menggunakan bahasa pemrograman PHP, HTML, CSS, dan Javascript dengan basis data *mySql*. Sistem Informasi Manajemen Akademik memberikan fitur-fitur sesuai dengan hasil analisis persyaratan. Hasil implementasi juga menghasilkan artefek sistem yang didokumentasikan

pada struktur artefak sitem, beberapa implementasi algoritme, dan beberapa implementasi antarmuka pengguna.

4. Hasil pengujian validasi pada beberapa fungsi yang berkaitan memasukkan data pendaftaran siswa baru dan mengelola data pokok siswa dinyatakan valid. Selanjutnya hasil pengujian kompatibilitas sistem terhadap peramban menunjukkan sistem berjalan dengan *Critical Issues* pada jenis peramban firefox versi 63 dan Chrome versi 70, Kemudian terdapat *Major issues* apabila situs diakses melalui peramban Internet Explorer versi 11 dan Andriod versi 4 atau sebelumnya.

## 8.2 Saran

Saran yang dapat diberikan sebagai bahan pertimbangan untuk melaksanakan pengembangan lanjut pada Sistem Informasi Manajemen Akademik pada Sekolah Dasar Negeri Dinoyo 1 adalah:

1. Perlu dilakukan penerapan lapangan Sistem Informasi Manajemen Akademik pada lokasi studi kasus Sekolah Dasar Negeri Dinoyo 1. Kemudian perlu dilakukan kegiatan evaluasi terhadap penerapan dari Sistem Informasi Manajemen Akademik sehingga memungkinkan hasil pengembangan lanjut yang lebih baik.
2. Perlu dilakukan evaluasi pada antarmuka pengguna Sistem Informasi Manajemen Akademik untuk mengetahui apabila perlu dilakukan perbaikan atau pengembangan lanjut antarmuka pengguna.

## DAFTAR REFERENSI

- 1keydata, 2009. *Physical Data Model*. [Online] Available at: <http://www.1keydata.com/datawarehousing/physical-data-model.html> [Diakses 18 Agustus 2018].
- Alshamrani, A., & Bahattab, A. (2015). A Comparison Between Three SDLC Models Waterfall Model , Spiral Model , and Incremental / Iterative Model, 12(1), 106–111.
- Becker, J., Kugeler, M. & Rosemann, M., 2013. *Process Management: A Guide for the Design of Business Processes*. Berlin: Springer-Verlag.
- Bittner, K. & Spence, I., 2002. *Use Case Modeling*. Boston: Addison Wesley.
- Booch, et. al., 2005. *The Unified Modeling Language User Guide*. 2nd Edition. United States : Pearson Education Inc.
- Firdausy, D. R., Wicaksono, S. A., & Pradana, F. (2018). Pengembangan Sistem Informasi Manajemen Pelaporan Sarana dan Prasarana Studi pada Fakultas Ilmu Komputer Universitas Brawijaya, 2(11), 5365–5374.
- IBM, (2004). DEV475 Mastering Object-Oriented Analysis and Design with. *International Business*, 3, 2–3.
- IBM, 2018. *Physical Data Models*. [Online] Tersedia di: [https://www.ibm.com/support/knowledgecenter/SS62YD\\_4.1.1/com.ibm.datatools.core.ui.doc/topics/cphysmod.html](https://www.ibm.com/support/knowledgecenter/SS62YD_4.1.1/com.ibm.datatools.core.ui.doc/topics/cphysmod.html) [Diakses 17 Agustus 2018].
- Laudon, K. & Laudon, J. P., 2011. *Management Information Systems*. 12 ed. s.l.:Pearson Education.
- Leffingwell, D. & Widrig, D., 2003. *Managing Software Requirements: A Use case Approach*. 2nd penyunt. Boston: Addison Wesley.
- Mishra, A. & Dubey, D., 2013. A Comparative Study of Different Software Development Life Cycle Models in Different Scenarios. *International Journal of Advance Research in Computer Science and Management Studies*, p. 6.
- Object Management Group, 2005. *OMG Unified Modeling Language Specification*. [Online] Available at: <http://www.omg.org/spec/UML/2.0/> [Diakses 17 Agustus 2018].
- Object Management Group, 2005. *Unified Modeling Language: Superstructure*. [Online] Available at: <http://www.omg.org/spec/UML/2.0/Superstructure/PDF/> [Diakses 17 Agustus 2018].
- Object Management Group, 2011. *Business Process Model and Notation (BPMN)*. [Online] Available at: <http://www.omg.org/spec/BPMN/2.0/PDF> [Diakses 17 Agustus 2018].

- O'Brien, J. A. & Marakas, G. M., 2013. Introduction to Information Systems. 16th Edition. United States.
- Oz, E., 2009. Management Information Systems. 6th ed. Massachusetts: Cengage Course Technology.
- Pitt, C., 2012. Pro PHP MVC. s.l.:Apress.
- Pressman, R. S., 2010. Software Engineering: A Practitioner's Approach. 7th penyunt. New York: McGraw-Hill.
- Przybyłek, A., 2011. *Bridging the gap between business process models and use-case models*. Gdańsk, Gdańsk University of Technology.
- Shah, M. (2013). Impact of Management Information Systems (MIS) on School Administration: What the Literature Says. *Procedia - Social and Behavioral Sciences*, 116, 2799–2804. <https://doi.org/10.1016/j.sbspro.2014.01.659>
- Shalahuddin, M. & S, R. A., 2015. Rekayasa Perangkat Lunak : Terstruktur dan Berorientasi Obyek. s.l.:Informatika.
- Sommerville, I., 2011. Software Engineering. 9th penyunt. Boston: Addison-Wesley.
- Suharjo, 2006. Mengenal Pendidikan Sekolah Dasar: Teori dan Praktek. Jakarta: Depdiknas.
- Visual Paradigm, 2016. *How To Draw Uml Sequence Diagram*. [online] Tersedia di: <https://www.visual-paradigm.com/tutorials/how-to-draw-uml-sequence-diagram.jsp> [Diakses 17 Agustus 2018]
- Wahyu, J., Faraday, A., Wicaksono, S. A., & Saputra, M. C. (2018). Pembangunan Sistem Informasi Manajemen Akademik Sekolah Dasar ( Studi Pada SDN Wates Kabupaten Kediri ), 2(11), 4777–4784.
- Weske, M., 2007. Business Process Management Concept, Languages, Architertures. New York: Springer-Verlag Berlin Heidelberg.